# Proyecto Fin de Carrera


# Sistema de Monitorización de la infraestructura CCTV en la UC3M con Zabbix

Alumno: Emilio Manuel González Pérez
Tutor: Juan Manuel Canelada Oset
Septiembre de 2010

*A todos los que han hecho esto posible.*

# AGRADECIMIENTOS

Estas páginas que véis aquí son las últimas páginas que redacto para completar la memoria de mi proyecto. Pero también son las primeras en las que, después de mucho tiempo, puedo escribir algo totalmente distinto.

No busquéis pues en estas palabras ninguna referencia a cámaras de CCTV, ni a sistemas de monitorización, ni a protocolos, ni a bibliografías… Lo único que encontraréis es a un tímido personaje al que le cuesta mucho trabajo hacer muestras públicas de agradecimiento. No soy ningún *Pepe Reina* hilando una broma tras otra, chiste tras chiste, agradeciendo a una nación entera el apoyo recibido. Puede que no haya ganado un Mundial ni una Eurocopa, pero, para mí, llegar hasta aquí es algo mucho más importante (que me perdonen los *futboleros*). También es posible que, en extensión, no tenga que dirigirme a todo un país, pero desde luego son muchas las personas a las que estoy agradecido, tantas que probablemente me deje a alguien en el tintero.

Empecemos entonces, y que nadie intente leer órdenes de importancia porque no los encontrará. A todos os tengo igual de presentes, lo prometo.

En primer lugar, quiero dar las gracias a mi tutor. Básicamente porque me ha aguantado todos estos meses durante los cuales ha dirigido mi proyecto y porque, siendo también un compañero de trabajo, tendrá que seguir soportándome durante un tiempo. Gracias por darme esta oportunidad, gracias por todo lo que me has inculcado, gracias por no '*haberme cortado el pelo a cuadros*' como tú dices cada vez que he metido la pata en algo, gracias por darme los empujones que necesitaba para llegar hasta este punto (si es por mí, se extingue el plan de estudios y yo sin acabar el proyecto), gracias por atenderme, gracias por seguir enseñándome cosas que ya debería tener bien aprendidas… Podría seguir, pero lo dejaré aquí. Si alguien quiere saber más sobre lo que significa tenerte como tutor, que se anime a pedirte un proyecto o un trabajo dirigido porque de verdad merece la pena.

Este proyecto nació como un objetivo dentro del que, desde hace tres años y medio, viene siendo mi lugar de trabajo. Cuando empecé mi andadura en el servicio de Informática de la Universidad Carlos III allá por el mes de febrero del año 2007, nunca pensé que llegaría a convertirse en prácticamente una segunda casa para mí. Extrañamente, y al contrario de lo que le ocurre a la mayoría de personas con su trabajo, he ido desarrollando un apego a este sitio hasta el punto de que no es raro verme por allí incluso en alguno de mis días de vacaciones. Estoy loco, lo sé. A todos mis jefes y no tan jefes del servicio de informática y del área de mantenimiento, a todos mis compañeros, a todos aquellos que estáis contentos con mi trabajo y a todos aquellos que pueden no estarlo… simplemente, gracias. Aunque no lo creáis, formar

parte de este servicio no sólo ha hecho que crezca profesionalmente sino que, a nivel personal, ha supuesto encontrarme con personas sin las que ahora, sinceramente, no sabría cómo vivir.

Y es precisamente a esas personas a las que, de ninguna manera, voy a pasar por alto. No es por el puro compromiso de seguir la línea de todos aquellos que se acuerdan de sus amigos en este tipo de situaciones. Realmente os quiero agradecer a todos los que considero amigos el haber estado ahí. Soy consciente de que no suelo demostraros a menudo el valor que tenéis para mí, sé que no soy quien más os llama por teléfono, sé que muchos de vosotros lleváis siglos sin verme, pero creedme cuando os digo que muchas de las fuerzas que me han llevado hasta aquí provienen de vosotros. A menudo os sorprende la buena memoria que tengo cuando de recordar cosas que tienen que ver con vosotros se trata. Citando a Tryon Edwards, "*el secreto de una buena memoria es la atención, y la atención a algo depende de nuestro interés en ello. Rara vez olvidamos algo que ha causado una profunda impresión en nosotros*". Si sigo recordando tantas cosas sobre vosotros es por la huella que dejáis en mí.

Por último, y no por ello menos importante, me gustaría hacer una mención especial a mi gente de casa, o familia como más gustéis. Consciente o inconscientemente, habéis ido moldeando a este chico que os escribe hasta convertirlo en lo que es hoy. Tal vez me equivoque, pero la sensación que tengo es que esperábais que, a estas alturas de mi vida, hubiera llegado más lejos. Por el momento, esto es lo que hay. Puede que sea poco, puede que sea mucho, puede que no sea nada comparado con todo lo que aún queda por llegar o con lo que podría haber llegado. En cualquier caso, vosotros sóis quienes me dísteis la oportunidad de formarme, quienes me pusísteis en el camino correcto para llegar a "terminar" mis estudios cuando menos motivación tenía, y entrecomillo esa palabra porque creo que esto no ha hecho más que empezar. Os debo muchísimo y creo que sería buena idea seguir trabajando cada día más para recompensaros de la mejor manera posible la dedicación que habéis tenido hacia mí.

# Resumen

Actualmente, la Universidad Carlos III de Madrid dispone de un sistema de videovigilancia con el objeto de complementar y mejorar los sistemas de seguridad y protección de personas, instalaciones y accesos existentes en los distintos campus de la misma. Dicho sistema también conocido como CCTV consiste en la captación, grabación y posterior consulta en cualquier momento por parte del personal autorizado de las imágenes almacenadas o en tiempo real, es pues un requisito imprescindible de este sistema la total disponibilidad del mismo tanto a la parte on-line como a las grabaciones almacenadas.

La solución tecnológica implantada, basada en cámaras IP y servidores con sistema operativo Windows necesitaba ser complementada con un sistema de gestión que permitiera a los administradores y responsables del mismo monitorizar y conocer en todo momento el estado de cada uno de los componentes (servidores de grabación, equipos de red, cámaras de vídeo, espacio de almacenamiento) de forma que se puedan detectar problemas antes de producirse, garantizando un nivel de servicio óptimo.

El presente proyecto tiene por objeto la implementación y despliegue de un sistema de monitorización de CCTV. Para ello, realiza un análisis en profundidad de la arquitectura y los parámetros a monitorizar del sistema CCTV implantado en la Universidad Carlos III de Madrid, extrayendo los requisitos fundamentales del mismo. Posteriormente se diseña, implementa y despliega una solución basada en Zabbix, herramienta OpenSource resultante de realizar una comparativa basada en casos de uso entre las principales herramientas de monitorización de sistemas existentes en la actualidad.

# Abstract

Nowadays, Carlos III University of Madrid (from now on, UC3M) has a video surveillance system which was deployed in order to complement and improve the existing security systems inside its facilities. Such system (also known as CCTV) bases its way of working on capturing and storing video information, so that authorized personnel can monitor the environment in real time or retrieve the recordings any time they want. Therefore, a total availability of that CCTV system (including recordings storage and on-line tier) is considered as a must.

The technical approach of the infraestructure of UC3M CCTV system is based on network cameras (IP monitoring) and dedicated servers with Windows installed as operating system. It needed to be complemented with a managing system that could help administrators and those in charge of security when it comes to monitoring at all times the status of each one of the components that give shape to the CCTV system (video recording servers, storage systems, and so on). That way, problems could be detected even before they took place and an optimal service level for the whole CCTV system would be guaranteed.

This project deals with the implementation and deployment of a CCTV monitoring system. To that end, a thorough analysis of the CCTV system in UC3M is conducted, focusing on existing architecture and parameters that should be monitored, and resulting in a list of essential requirements. After that, a Zabbix-based solution will be designed, implemented and deployed. Zabbix is an *OpenSource* monitoring tool which is chosen as a result of a comparison between it and two of the most popular existing monitoring tools.

# CONTENIDO

# ÍNDICE DE TABLAS

# ÍNDICE DE IMÁGENES

# 1

# Introducción

# 1. INTRODUCCIÓN

El presente documento explica el proyecto de instalación de un sistema de monitorización de una infraestructura de videovigilancia en un ámbito concreto como es la Universidad Carlos III de Madrid (UC3M). Se explicarán los detalles del trabajo seguido hasta lograr la adaptación de una herramienta que permita cubrir las particulares necesidades del entorno de videovigilancia que nos ocupa.

## 1.1. Motivación

En la sociedad actual no son pocos los sucesos que han supuesto un replanteamiento en lo que al concepto de seguridad se refiere. El ciudadano vive en medio de un ambiente en el que se encuentra relativamente vulnerable, desprovisto de protección. Sentimos que no estamos suficientemente seguros, y esa sensación nos conduce a pensar que, aunque perdamos ciertas libertades, podemos asumir ciertos sacrificios por el bien de nuestra propia seguridad.

Entre esos sacrificios se encuentra la instalación de cámaras de circuito cerrado (CCTV[1]), cada vez más extendidas en los espacios públicos, espacios donde el concepto de seguridad cobra más importancia.

La Universidad Carlos III de Madrid no es ajena a este propósito de gestión de la seguridad y, actualmente, dispone de circuitos cerrados de televisión instalados en sus campus de Getafe y de Leganés. La línea que sigue la implementación del sistema en ambos campus se basa en una serie de cámaras conectadas a una infraestructura de red privada a la que se encuentra conectado un conjunto de servidores, siendo éstos los encargados de almacenar las grabaciones que reciben desde las cámaras. Por otra parte, como en todo sistema de videovigilancia existente, las imágenes que llegan en tiempo real de las cámaras así como las grabaciones existentes son monitorizadas desde equipos manipulados por las entidades correspondientes.

Todo sistema informático requiere unas labores de gestión y monitorización, si bien en el caso de los sistemas de videovigilancia hay necesidades más acentuadas en aspectos como la **disponibilidad**. Éste es un factor crítico, pues el sistema de videovigilancia siempre tiene que estar en servicio, funcionando ininterrumpidamente, las veinticuatro horas del día, durante siete días a la semana y todo ello durante los doce meses de que consta un año. Todo esto supondrá unas necesidades especiales de monitorización que nos ayuden a conseguir esa alta disponibilidad del sistema.

Otro punto a considerar, común a gran parte de otros sistemas, son las restricciones de acceso que suponen las condiciones de instalación del sistema de

[1] CCTV, acrónimo de "Circuito Cerrado de Televisión"

videovigilancia. Tal como se mencionó en un párrafo anterior, el sistema de videovigilancia tiene sus cimientos, en cuanto a arquitectura de conexiones se refiere, sobre una red privada, independiente de la infraestructura de red del resto de la Universidad.

Tradicionalmente, desde el punto de vista físico existen dos vertientes en lo que a monitorización de un sistema se refiere: local o remotamente. Si afrontamos la monitorización del sistema de videovigilancia desde el punto de vista local, nos encontramos con las primeras desventajas de esta metodología. La primera es la necesidad de desplazarse al lugar en el que se encuentran instalados físicamente los servidores de grabación del circuito de televisión, pues en éstos se encuentran instaladas herramientas de diagnóstico y otro tipo de software gracias a los cuales podemos saber si las cámaras y el sistema de almacenamiento de grabaciones están funcionando correctamente. En el sistema de videovigilancia hay un factor especialmente crítico del lado de los servidores de grabación, y éste no es otro que el espacio de almacenamiento. Como ya hemos dicho, las grabaciones de las imágenes obtenidas por las cámaras de videovigilancia son almacenadas en los servidores, por lo que en éstos se debe vigilar especialmente que haya espacio donde guardarlas.

¿Qué nos supondría vigilar factores tan críticos como el espacio en disco duro? La experiencia nos dice que las visitas a los centros donde se encuentran instalados los servidores se hacen totalmente imprescindibles y considerablemente frecuentes, lo cual consume esfuerzo tanto en tiempo como en recursos. Si deseamos ahorrarnos de algún modo o facilitar ese trabajo que supone el tener que moverse hasta los servidores y optamos por la alternativa de conectarnos remotamente a ellos, deberemos hacerlo desde un equipo conectado a la red privada que señalábamos anteriormente y ésta es una estrategia que se debe tener bien controlada dadas las especiales características de dicha red.

Una de las primeras soluciones de monitorización es un simple script, invocado regularmente (p.ej. a través de un crontab[2]) para comprobar parámetros básicos del sistema, como pueden ser el espacio en disco o el estado de determinados servicios. Conforme crece el número de equipos en el sistema y de parámetros a monitorizar en los mismos, ese script inicial va convirtiéndose en una solución cuyo mantenimiento absorbe más tiempo del que realmente se intenta ahorrar con la creación de dicho script.

En una situación como la descrita, desprovistos de una herramienta de monitorización, si bien es cierto que puede llevarse a cabo una supervisión medianamente aceptable del funcionamiento del sistema, perdemos un componente

[2] Programa que permite ejecutar otros programas o scripts en un lapso de tiempo y una periodicidad especificada por el usuario.

esencial en la mayoría de los sistemas informáticos, y éste no es otro que la **automatización**.

Por este motivo surge la principal motivación del proyecto, la de detectar los problemas de forma proactiva, poder anticiparnos a los posibles incidentes que pudiesen surgir, como intrusiones, falta de recursos… y, por encima de todo, poder garantizar el nivel de servicio deseado para el sistema CCTV en la UC3M.

Con objeto de cubrir estas necesidades, el requisito principal se resume en implantar una plataforma basada en un sistema de monitorización por medio del cual vigilar, supervisar y actuar frente a esos posibles incidentes que podrían arruinar la disponibilidad del sistema, todo ello de la forma más automática posible.

Adicionalmente, se precisará poder contar con un histórico de datos sobre el comportamiento del sistema para poder llevar a cabo una labor de seguimiento que nos ayude en la línea de incorporación de mejoras en la infraestructura del sistema a nivel general.

## 1.2. Objetivos

El objetivo principal se resume en la implantación de una plataforma con la que monitorizar la infraestructura del sistema de videovigilancia de la UC3M, utilizando para ello la herramienta de monitorización Zabbix. A través de Zabbix se supervisará el buen funcionamiento de todos los servidores de grabación, las cámaras de videovigilancia, la electrónica de red y los equipos de los centros de control, recogiendo datos sobre su comportamiento para posteriormente guardarlos en una Base de Datos MySQL.

La monitorización llevada a cabo en la plataforma será tanto a nivel de *hardware* (espacio en disco, porcentaje de uso de CPU y de memoria, procesos, etc.) como a nivel de *software*, en cuyo caso se prestará especial atención al control del funcionamiento de la aplicación propietaria que gestiona la grabación de las imágenes de vídeo recogidas por las cámaras (*Sony RealShot Manager*), así como al resto de servicios necesarios para la toma de datos sobre el rendimiento del sistema (agente SNMP, etc.).

A la hora de monitorizar los equipos seguiremos una estrategia de división en grupos compuestos por equipos cuyas características sean homogéneas entre sí. De esta forma, tendremos grupos diferenciados para los servidores de grabación, las cámaras de videovigilancia, los centros de control de los dos campus y los distintos tipos de elementos de la electrónica de red. Analizaremos las necesidades de monitorización específicas para cada uno de esos grupos en lugar de establecer parámetros de monitorización para cada equipo, simplificando así el trabajo que llevaremos a cabo en Zabbix.

## 1.3. Estructura del documento

Este documento está compuesto de una serie de capítulos estructurados en la forma que sigue a continuación:

1. **Introducción**: el capítulo de introducción supone una primera toma de contacto con el contexto dentro del cual se encuadra el proyecto. Básicamente incluye una descripción del propósito que motiva el trabajo llevado a cabo en la realización de este proyecto de fin de carrera.

2. **Gestión del proyecto:** se incluyen en esta sección la planificación inicial de las tareas a desempeñar en el desarrollo de la plataforma de monitorización así como una estimación de los costes asociados al proyecto en sí mismo.

3. **Estado de la cuestión**: en este apartado se describe la situación actual en lo que respecta al concepto de sistema de monitorización y a los aspectos relacionados con éste. Se hace una presentación de las soluciones de monitorización estudiadas así como de conceptos relacionados haciendo especial hincapié en la herramienta finalmente escogida para implementar la plataforma de monitorización del sistema de videovigilancia.

4. **Análisis del sistema de videovigilancia**: en este punto se estudia el entorno en el que se desplegará la plataforma de monitorización, las necesidades a cubrir y los requisitos que debe satisfacer la solución que finalmente se adopte**.**

5. **Diseño de la plataforma de monitorización:** en esta etapa de diseño se plantean varias soluciones software que cumplan los requisitos propuestos en la fase de análisis. Se incluye una comparativa entre esas soluciones así como una justificación que explique el porqué de la herramienta finalmente escogida. Asimismo, se incluye aquí una descripción del diseño de la plataforma a la que dicha herramienta servirá de base.

6. **Despliegue de la plataforma de monitorización:** el despliegue se refiere a la implementación propiamente dicha de la plataforma**.** Aquí se explicarán todos los pasos seguidos para dar forma a las decisiones de diseño tomadas en el apartado anterior.

7. **Plan de pruebas**: se incluye en este apartado un plan con las pruebas más representativas llevadas a cabo para verificar la monitorización del sistema CCTV. Especificaremos pruebas unitarias, pruebas de integración y pruebas de aceptación.

8. **Plan de mantenimiento:** con la plataforma ya desplegada, verificada y aceptada, se confecciona un plan de mantenimiento a niveles preventivo, correctivo y adaptativo con objeto de garantizar que el sistema de monitorización así como las demás partes involucradas se encuentran en todo momento en un estado óptimo de funcionamiento.

9. **Plan de contingencia**: en este punto se explicará cómo se llevan a cabo las copias de seguridad tanto del equipo en el que se instala Zabbix como de la base de datos en la que se registran los valores obtenidos en el proceso de monitorización.

10. **Conclusiones y líneas futuras de desarrollo:** se comentan las conclusiones a la que se llega tras completar las distintas fases de que se compone el proyecto y se exponen los trabajos futuros que podrían seguirse en la línea de monitorización del sistema de videovigilancia.

11. **Bibliografía:** en este apartado se indica el material bibliográfico consultado para la realización del proyecto.

12. **Definiciones y acrónimos:** se incluirán aquí los principales conceptos y siglas que hayan sido referenciados en el presente documento.

13. **Referencias:** se recogen aquí las referencias utilizadas en el desarrollo del proyecto y en la redacción del presente documento.

14. **Anexos:** recogeremos aquí información que aporte contenido adicional al proyecto actual.

# 2

# Gestión del proyecto

# 2. GESTIÓN DEL PROYECTO

Todo proyecto desarrollado desde la perspectiva de la ingeniería debe contar con un proceso de gestión que incluya una planificación y una organización con las que ayudar a cumplir los objetivos y a satisfacer en un tiempo, coste y recursos determinados las necesidades que motivaron la concepción del proyecto.

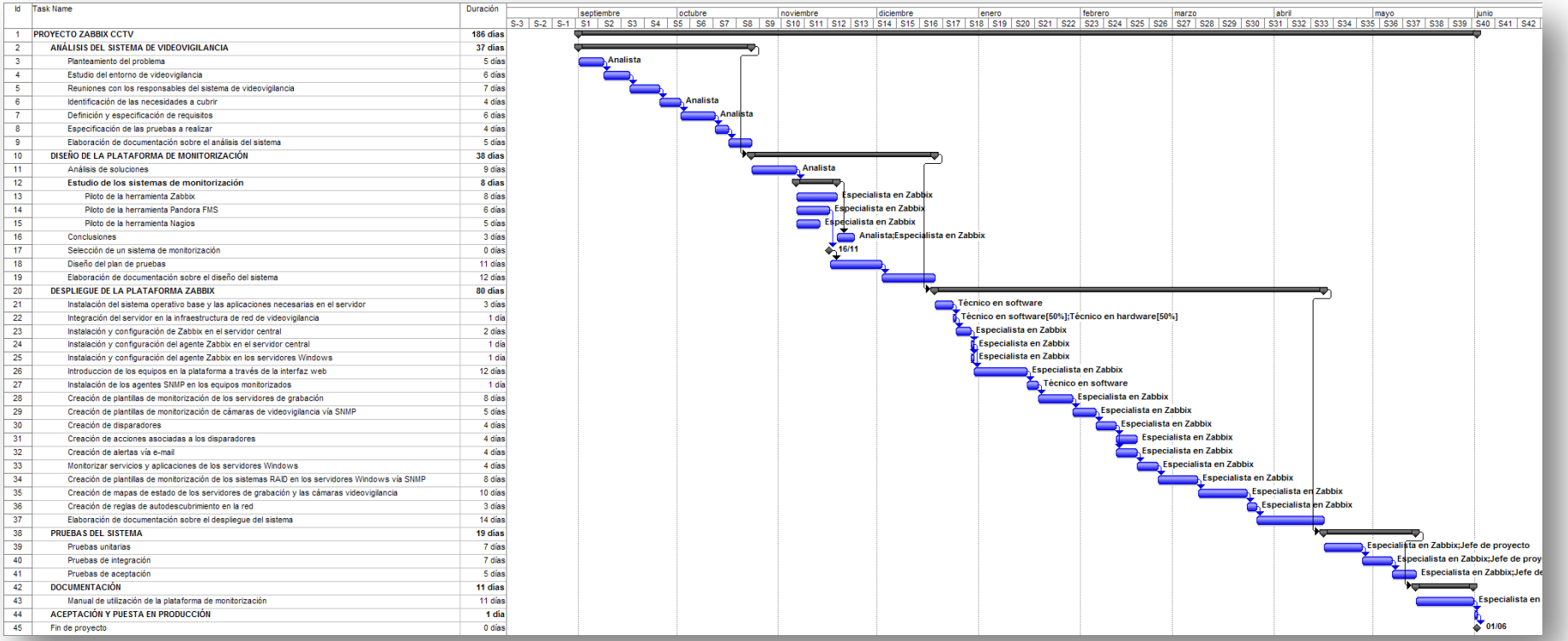
En este capítulo se presentan los procedimientos llevados a cabo en la gestión del proyecto que nos ocupa. Se definirá en primer lugar la planificación inicial de tareas básicas del proyecto así como las actividades realmente importantes que nos llevarán a la consecución de un proyecto que satisfaga las necesidades establecidas.

Veremos referencias a la organización seguida a lo largo del proyecto así como un análisis de los costes que el desarrollo del mismo ha supuesto para el entorno involucrado.

## 2.1. Planificación inicial

El inicio del proyecto que nos ocupa data del mes de septiembre del año 2009, momento en el cual se procedió a realizar una planificación inicial en la que figuraran las tareas que constituirían el desarrollo del proyecto.

Para verlo más claramente, el diagrama de Gantt que sigue nos presenta las tareas llevadas a cabo con el coste temporal asociado a cada una de ellas.



| Id | Task Name | Duración |
|---|---|---|
| 1 | **PROYECTO ZABBIX CCTV** | **186 días** |
| 2 | **ANÁLISIS DEL SISTEMA DE VIDEOVIGILANCIA** | **37 días** |
| 3 | Planteamiento del problema | 5 días |
| 4 | Estudio del entorno de videovigilancia | 6 días |
| 5 | Reuniones con los responsables del sistema de videovigilancia | 7 días |
| 6 | Identificación de las necesidades a cubrir | 4 días |
| 7 | Definición y especificación de requisitos | 6 días |
| 8 | Especificación de las pruebas a realizar | 4 días |
| 9 | Elaboración de documentación sobre el análisis del sistema | 5 días |
| 10 | **DISEÑO DE LA PLATAFORMA DE MONITORIZACIÓN** | **38 días** |
| 11 | Análisis de soluciones | 9 días |
| 12 | **Estudio de los sistemas de monitorización** | **8 días** |
| 13 | Piloto de la herramienta Zabbix | 8 días |
| 14 | Piloto de la herramienta Pandora FMS | 6 días |
| 15 | Piloto de la herramienta Nagios | 5 días |
| 16 | Conclusiones | 3 días |
| 17 | Selección de un sistema de monitorización | 0 días |
| 18 | Diseño del plan de pruebas | 11 días |
| 19 | Elaboración de documentación sobre el diseño del sistema | 12 días |
| 20 | **DESPLIEGUE DE LA PLATAFORMA ZABBIX** | **80 días** |
| 21 | Instalación del sistema operativo base y las aplicaciones necesarias en el servidor | 3 días |
| 22 | Integración del servidor en la infraestructura de red de videovigilancia | 1 día |
| 23 | Instalación y configuración de Zabbix en el servidor central | 2 días |
| 24 | Instalación y configuración del agente Zabbix en el servidor central | 1 día |
| 25 | Instalación y configuración del agente Zabbix en los servidores Windows | 1 día |
| 26 | Introduccion de los equipos en la plataforma a través de la interfaz web | 12 días |
| 27 | Instalación de los agentes SNMP en los equipos monitorizados | 1 día |
| 28 | Creación de plantillas de monitorización de los servidores de grabación | 8 días |
| 29 | Creación de plantillas de monitorización de cámaras de videovigilancia vía SNMP | 5 días |
| 30 | Creación de disparadores | 4 días |
| 31 | Creación de acciones asociadas a los disparadores | 4 días |
| 32 | Creación de alertas vía e-mail | 4 días |
| 33 | Monitorizar servicios y aplicaciones de los servidores Windows | 4 días |
| 34 | Creación de plantillas de monitorización de los sistemas RAID en los servidores Windows vía SNMP | 8 días |
| 35 | Creación de mapas de estado de los servidores de grabación y las cámaras videovigilancia | 10 días |
| 36 | Creación de reglas de autodescubrimiento en la red | 3 días |
| 37 | Elaboración de documentación sobre el despliegue del sistema | 14 días |
| 38 | **PRUEBAS DEL SISTEMA** | **19 días** |
| 39 | Pruebas unitarias | 7 días |
| 40 | Pruebas de integración | 7 días |
| 41 | Pruebas de aceptación | 5 días |
| 42 | **DOCUMENTACIÓN** | **11 días** |
| 43 | Manual de utilización de la plataforma de monitorización | 11 días |
| 44 | **ACEPTACIÓN Y PUESTA EN PRODUCCIÓN** | **1 día** |
| 45 | Fin de proyecto | 0 días |



**Figura 1. Planificación inicial del proyecto**

La planificación inicial estima **186 días** para concluir el proyecto con una dedicación definida de unas **7 horas diarias** durante **22 días al mes**. Así, se tiene un resultado de **1302 horas** que tomaría inicialmente el desarrollo del proyecto.

## 2.2. Organización

El éxito de todo proyecto depende en gran medida de la calidad seguida en el proceso de desarrollo, especialmente cuando éste implica sistemas informáticos, en cuyo caso el mantenimiento de éstos también es parte fundamental del ciclo de vida.

La metodología a seguir se estructura de manera que conduzca a mejorar el proceso de desarrollo, y, para ello, en este caso particular, adopta un enfoque iterativo consistente en tres fases que siguen un proceso cíclico. Estas tres etapas serían las siguientes:

- **Análisis del sistema de videovigilancia**: inicialmente se procede a un análisis de las necesidades del sistema que conforma la infraestructura de videovigilancia instalada. La idea es llegar a identificar **Qué** se desea conseguir con el proyecto. Siguiendo una serie de reuniones y conversaciones con los responsables del área de seguridad del servicio de informática y comunicaciones de la UC3M así como los responsables del área de seguridad física, se establecen las bases para definir las necesidades que se desean cubrir y los objetivos que se persiguen alcanzar a lo largo del proyecto. Dentro de esta fase se especificarán las pruebas que llevarán a cabo para verificar el correcto funcionamiento del sistema de monitorización a implementar y, finalmente, como resultado de esta fase, se elaborará la documentación que refleje la actividad seguida.

- **Diseño del sistema de monitorización**: el diseño constituye una aproximación a la resolución del problema planteado, al **Cómo** lo solucionaremos. Una vez definidas las necesidades, se procede al estudio de éstas para establecer un diseño a partir del cual construir una solución capaz de satisfacer esos requisitos. Se concretan aquí las decisiones en cuanto a la arquitectura de la plataforma así como los elementos a implantar en la misma. Igualmente, se definirá el plan de pruebas en el que se incluirán las distintas pruebas extraídas en la fase de análisis. Al igual que la fase de análisis, se redactará la documentación necesaria que incluya las decisiones de diseño.

- **Despliegue de la plataforma de monitorización**: en este punto se procede a la construcción o implementación de la plataforma basada en Zabbix de acuerdo a las decisiones de diseño adoptadas en la fase anterior y que satisfacen los requisitos definidos en la etapa de análisis. Se llevarán a cabo tareas a nivel hardware, como la instalación del servidor en el cual se alojará la plataforma de monitorización. A nivel software, se incluye la instalación de la herramienta Zabbix y el despliegue en los servidores de grabación y equipos de los centros de control, para lo cual se instalarán en ellos el agente Zabbix y el agente SNMP. En el lado de las cámaras de videovigilancia y los equipos de la electrónica de red, el despliegue supondrá la configuración del agente SNMP

correspondiente. Todas estas acciones se detallarán en la correspondiente documentación de esta fase.

- **Pruebas del sistema:** una vez concluido el proceso de construcción de la plataforma, se establece un plan de pruebas a distintos niveles (pruebas unitarias y pruebas de aceptación) para comprobar la completa funcionalidad del sistema, garantizando así que cada uno de sus componentes está libre de fallos. Una vez superadas las pruebas definidas, se procederá a la aceptación final y puesta en producción de la plataforma creada.

- **Documentación:** junto con la documentación presentada en las fases de análisis, diseño y despliegue, y posteriormente a la aceptación y puesta en producción de la plataforma, se redactará un manual de utilización de la plataforma con el que proveer asistencia técnica a cualquier usuario de la misma.

- **Mantenimiento**: dada la especial criticidad del sistema de videovigilancia, se considera totalmente necesario no sólo implementar una plataforma de monitorización, sino también concebir un plan de mantenimiento de la misma a niveles preventivo (ajustes de rendimiento), correctivo (solución de fallos encontrados) y evolutivo (estudio e introducción de mejoras que se ajusten a nuevas necesidades descubiertas).

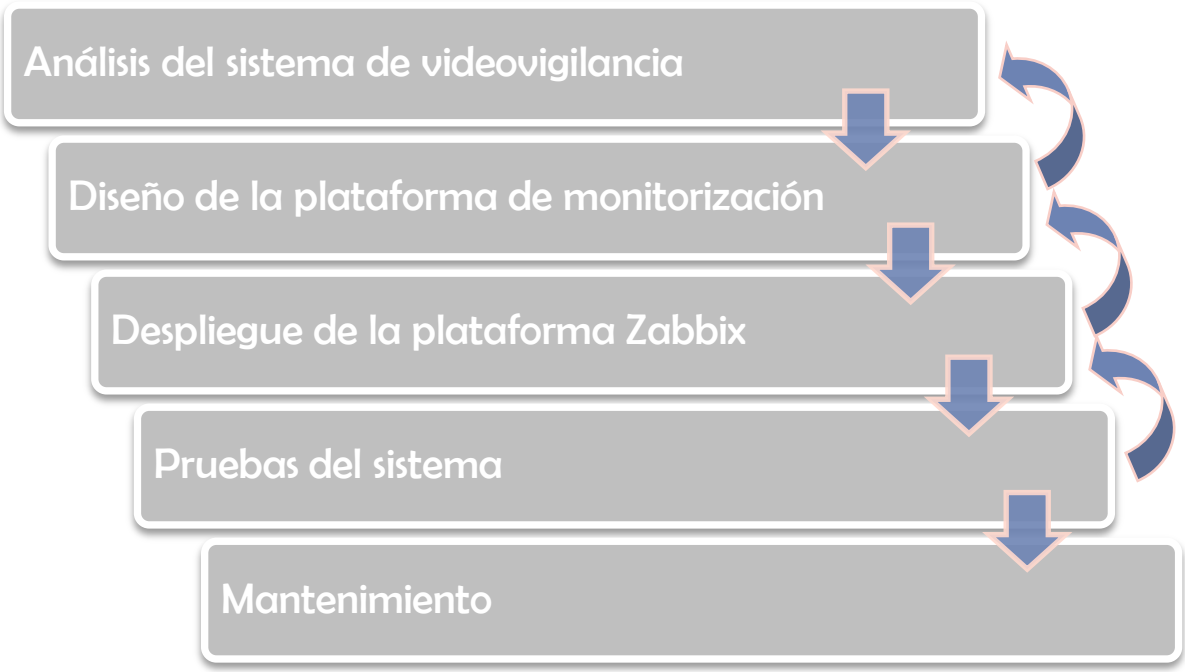



**Figura 2. Diagrama del proceso de desarrollo del proyecto**

## 2.3. Análisis de costes

En este capítulo veremos una estimación de los costes y del presupuesto que supone la implementación del proyecto. Nótese que se trata de una estimación simulada o ficticia y que, por tanto, aunque se manejen cifras económicas, no se exigirá a ninguna entidad en ningún momento el pago de los costes en ese apartado representados.

El coste del proyecto se estima en euros (€) y se calcula en base a la planificación y los recursos, ya sean humanos o técnicos, necesarios para llevar a cabo el mismo. Se definirá una estrategia de cálculo de costes en función de la planificación inicial establecida en apartados anteriores.

### 2.3.1. Duración del proyecto

En este punto se indica el coste en horas que supone la consecución de cada una de las tareas detalladas en la planificación inicial del proyecto.

| Tarea | Duración |
|---|---|
| Análisis del sistema de videovigilancia | 37 días * 7 horas/día = 259 horas |
| Diseño de la plataforma de monitorización | 38 días * 7 horas/día = 266 horas |
| Despliegue de la plataforma Zabbix | 80 días * 7 horas/día = 560 horas |
| Pruebas del sistema | 19 días * 7 horas/día = 133 horas |
| Documentación | 11 días * 7 horas/día = 77 horas |
| **Total** | **1295 horas** |

**Tabla 1. Duración de las tareas del proyecto**

## 2.3.2. Estrategia de cálculo de costes

Tradicionalmente, los costes asociados a un proyecto siguen una línea de división en dos grandes grupos: costes directos y costes indirectos. Por costes directos entendemos aquellos costes directamente asociados con la implementación del proyecto en sí mismo, y consideraremos como costes indirectos a todos los costes que son relativamente independientes del proceso de desarrollo.

Como costes directos podemos señalar los costes que suponen los diversos recursos humanos asignados a las tareas definidas en la planificación inicial, así como los costes hardware y software. Ejemplos de costes indirectos pueden ser los gastos derivados de comunicaciones, electricidad, etc.

En el caso de los costes indirectos, habitualmente se incluyen como tales el gasto en agua, en electricidad, en comunicaciones, etc. Dado el carácter de este proyecto, no consideraremos el cálculo de los costes indirectos.

### Cálculo de costes directos

Para llevar a cabo este cálculo consideraremos los costes relacionados con los recursos humanos involucrados en el desempeño de las distintas fases del proyecto y los costes que suponen el hardware y software utilizados.

#### A) Costes de Recursos Humanos

Cuando se define una planificación, se especifica no sólo el conjunto o listado de tareas con el trabajo a realizar, sino también los recursos, tanto materiales como humanos, que se encargarán de que ese trabajo se complete.

Se ha definido una serie de roles que juegan el papel de los distintos empleados que participan en el proyecto, procurando abarcar en lo posible los distintos rangos profesionales que habitualmente desempeñan las tareas de las que consta la planificación del proyecto.

Los roles son los siguientes:

- **Analista**: el analista es especialmente importante por su peso en la especificación de las necesidades a cubrir y en la definición del plan de pruebas que lleve a la aceptación final del proyecto.
- **Jefe de proyecto**: toma parte en tareas fundamentales como la documentación y los hitos del proyecto.
- **Técnico en hardware**: necesario para el montaje de la infraestructura física del sistema de videovigilancia.
- **Técnico en software**: se encarga de la instalación de las aplicaciones básicas necesarias para que funcione la plataforma de monitorización.
- **Especialista en Zabbix**: lleva la mayor parte del peso del proyecto, al tener a su cargo el núcleo fundamental de las tareas a desarrollar en la construcción de la plataforma.

Para el cálculo que implica disponer de estos recursos humanos se tiene en cuenta el salario de cada uno de ellos. Orientativamente, para una jornada laboral de 7 horas diarias durante 22 días al mes, los salarios quedan representados en la siguiente tabla:

| Categoría profesional | Sueldo Bruto/Año | Sueldo Bruto/Mes | Coste Hora/Empleado |
|---|---|---|---|
| Analista | 30.000 € | 2.500 € | 16 € |
| Especialista en Zabbix | 25.000 € | 2.000 € | 13 € |
| Jefe de proyecto | 40.000 € | 3.000 | 20 € |
| Técnico en hardware | 20.000 € | 1.500 € | 10 € |
| Técnico en software | 20.000€ | 1.500 € | 10 € |

**Tabla 2. Costes según rol de desarrollo**

Ahora que ya conocemos el salario del personal que lleva cabo las tareas y teniendo en cuenta la duración de cada una de esas tareas en las que participan, podemos proceder al cálculo del coste que todo ello supondría:

| Tarea | Duración | Recurso | Coste unitario | Total |
|---|---|---|---|---|
| **Análisis del sistema de videovigilancia** | 259 h | Analista (100%) | 16 €/h | 259 h * 16 €/h = 4144 €<br><br>**Total = 4144 €** |
| **Diseño de la plataforma de monitorización** | 266 h | Analista (30%) | 16 € /h | 0,3 * (16 €/h) * 266 h = 1276,8 € |
| | | Especialista en Zabbix (70%) | 13 € /h | 0,7 * (13 €/h) * 266 h = 2420,6 €<br><br>**Total = 3697,4 €** |

| | | | | |
|---|---|---|---|---|
| **Despliegue de la plataforma Zabbix** | 560 h | Técnico en software (15%) | 10 €/h | 0,15 * (10 €/h) * 560 h = 840 € |
| | | Técnico en hardware (5%) | 10 €/h | 0,05 * (10 €/h) * 560 h = 280 € |
| | | Especialista en Zabbix (80%) | 13 €/h | 0,8 * (13 €/h) * 560 h = 5824 € |
| | | | | **Total = 6944 €** |
| **Pruebas del sistema** | 133 h | Analista (50%) | 16 € /h | 0,5 * (16 €/h) * 133 h = 1064 € |
| | | Especialista en Zabbix (50%) | 13 € /h | 0,5 * (13 €/h) * 133 h = 864,5 € |
| | | | | **Total = 1928,5 €** |
| **Documentación** | | Especialista en Zabbix (100%) | 13,25€/h | 13 €/h * 77 h = **1001 €** |
| | | | | **Total = 1001 €** |
| **Coste total** | | | | **17714,9 €** |

**Tabla 3. Cálculo del coste de Recursos Humanos**

## B) Costes de Hardware

Las herramientas hardware utilizadas así como el coste de las mismas quedan representadas en la siguiente tabla:

| Cantidad | Material | Coste |
|---|---|---|
| 1 | Servidor HP Proliant DL 360 G2 | **2600 €** |
| 2 | Discos duros SCSI 300 GB | **750 €** |
| **Total** | | **3350 €** |

**Tabla 4. Cálculo del coste de Hardware**

## C) Costes de Software

Los recursos software de los que se ha hecho tienen asociado un coste tal como describe la tabla que sigue:

| Cantidad | Material | Coste |
|---|---|---|
| 1 | Licencia Ubuntu Server 9.04 | 0,00 € |
| 1 | Licencia servidor Zabbix | 0,00 € |
| 12 | Licencia agentes Zabbix | 0,00 € |
| 12 | Agente SNMP 3ware | 0.00 € |
| 1 | Licencia servidor MySQL | 0,00 € |
| 1 | Licencia servidor Apache | 0,00 € |
| **Total** | | **0,00 €** |

**Tabla 5. Cálculo del coste de Software**

## 2.3.3. Estimación inicial de costes

Partiendo de los costes temporales y de los costes, tanto directos como indirectos, del apartado anterior se elabora la siguiente tabla:

**NOTA:** representaremos los recursos humanos como **RRHH**, los recursos software como **SW** y los recursos hardware como **HW**.

| Recurso | Concepto | Coste |
|---|---|---|
| **RRHH** | **Análisis del sistema de videovigilancia** | 4144 € |
| **RRHH** | **Diseño del sistema de monitorización** | 3697,4 € |
| **RRHH** | **Despliegue de la plataforma Zabbix** | 6944 € |
| **RRHH** | **Plan de pruebas** | 1928,5 € |
| **RRHH** | **Documentación** | 1001 € |
| **HW** | **Servidor HP Proliant DL 360 G2** | 2600 € |
| **HW** | **Discos duros SCSI 300 GB** | 750 € |
| **SW** | **Licencia Ubuntu Server 9.04** | 0,00 € |
| **SW** | **Licencia servidor Zabbix** | 0,00 € |
| **SW** | **Licencia servidor Pandora** | 0,00 € |
| **SW** | **Licencia agentes Pandora** | 0,00 € |
| **SW** | **Agente SNMP 3ware** | 0,00 € |
| **SW** | **Licencia servidor MySQL** | 0,00 € |
| **SW** | **Licencia servidor Apache** | 0,00 € |
| **Total** | | **21604,9 €** |

**Tabla 6. Estimación de costes para el proyecto**

A la vista de la tabla anterior, obtenemos una estimación de costes que asciende a **21604,9** € para llevar a cabo el proyecto en los plazos de tiempo (**9 meses**) obtenidos en la planificación inicial.

# 3

# Estado de la cuestión

# 3. ESTADO DE LA CUESTIÓN

Este capítulo nos permitirá conocer los conceptos teóricos más importantes bajo los que se asienta la plataforma de monitorización a desarrollar.

## 3.1. ¿Por qué un sistema de monitorización?

Imagine por un momento que usted, administrador de un sistema, recibe de repente en su teléfono móvil una llamada informándole de que uno de los servidores que administra se ha caído y necesita estar de nuevo en funcionamiento antes de la mañana del día siguiente. Imagine que se dirige a su oficina para ver in situ el problema y descubre que, simplemente, algunos ficheros de log han crecido más de la cuenta durante la pasada semana y han terminado por agotar la totalidad del espacio disponible en el disco duro.

Aunque el escenario descrito sea bastante simplista, es algo que podría sucederle a cualquier administrador de sistemas. Para evitarlo, una de las opciones es buscar una solución con la que monitorizar su hardware de red, sus servidores, la disponibilidad de su web, etc. A día de hoy, las herramientas de monitorización que supervisan información de la red y de los equipos instalados en ella son prácticamente algo imprescindible y su importancia es algo que crece exponencialmente con el número de equipos a administrar y mantener.

Aquí es donde entra en juego Zabbix, una de las tantas herramientas de monitorización con las que poder llevar a cabo esa importantísima labor de supervisión de, en este caso, un sistema tan crítico como el sistema de videovigilancia de la UC3M.

## 3.2. Zabbix

Dentro de este capítulo se detallan las características más notables de la herramienta de monitorización Zabbix a la hora de construir la plataforma de monitorización del sistema que nos ocupa.

### 3.2.1. ¿Qué es Zabbix?

Zabbix es una solución de monitorización Open Source creada por Alexei Vladishev totalmente gratuita, escrita y divulgada bajo la licencia GPL[3], por lo que su código fuente está disponible para todos los usuarios finales y es distribuido sin coste alguno.

[3] General Public License

### 3.2.2. Qué ofrece Zabbix

Zabbix proporciona diversas maneras de monitorizar diferentes aspectos en una infraestructura tecnológica dada. Se puede caracterizar Zabbix como un sistema de monitorización semi-distribuido con gestión centralizada. Mientras la mayoría de instalaciones tienen una única Base de Datos centralizada, Zabbix posibilita monitorización distribuida a través de nodos y proxies.

Las características que podemos encontrar en Zabbix se resumen en la siguiente lista:

- Interfaz web centralizada y fácil de utilizar.
- Servidor que funciona bajo la mayoría de sistemas operativos basados en Unix, incluyendo Linux, AIX, FreeBSD, OpenBSD y Solaris.
- Agentes nativos para la práctica mayoría de los anteriores sistemas basados en Unix y para las versiones de Windows.
- Desarrollo de gráficos integrado y diversas prestaciones de visualización.
- Notificaciones que permiten una fácil integración con otros sistemas.
- Envío de alertas vía e-mail, SMS y servicios de mensajería instantánea.
- Ejecución de comandos remotos desde el servidor central.
- Configuración flexible incluyendo la definición de plantillas.

### 3.2.3. Cómo funciona Zabbix

La arquitectura más comúnmente utilizada en los sistemas que hacen uso de Zabbix es una arquitectura **Agente-Servidor**, basada en un servidor que obtiene los datos a través de consultas ejecutadas sobre una serie de agentes. Un agente es simplemente un programa escrito en C e instalado en un equipo (en adelante nos referiremos a cada uno de esos equipos como "host") que se desea monitorizar. Un host puede ser, por ejemplo, un servidor. En general, podremos catalogar como host cualquier dispositivo que sea identificable en una infraestructura de red, como puede ser una cámara de videovigilancia en nuestro caso.

En cualquier caso, no siempre es un requisito indispensable disponer de un agente instalado en el host a monitorizar. De hecho, nos encontraremos con equipos en los que, dadas sus características, no sea posible la instalación del agente. En esos casos Zabbix puede llevar a cabo comprobaciones sencillas sobre el equipo, como puede ser un ping o la comprobación de servicios http o ftp a través de su puerto establecido.

Sin embargo, si realmente se quieren explotar las posibilidades de monitorización que Zabbix ofrece, lo ideal es tener instalado un agente en el host siempre que sea posible, pues de esta manera se podrán recopilar y tratar muchos más datos con los

que obtener una información mucho más fiable. Una vez instalado, el agente funciona como un servicio más en el host y, a través de llamadas al sistema, recopila la información que se necesita para el proceso de monitorización desde el servidor principal.

La comunicación entre agente y servidor es sencilla, y se realiza a través de los puertos 10050 y 10051, ambos TCP. El host donde está instalado el agente "escucha" las peticiones que el servidor envía, y lo hace a través del puerto 10050. Por su parte, una vez recopilados los datos por el agente, el servidor recibe éstos por su puerto 10051 local.

Una vez definida la arquitectura "interna" sobre la que sea asienta el funcionamiento de Zabbix, es turno de explicar cómo el usuario final se comunica con la herramienta en los procesos de configuración, visualización de datos, etc.

Zabbix dispone de un "**frontend**" o interfaz web, escrito en PHP, que nos proporciona varias opciones para visualizar los datos recogidos, desde listas de problemas y gráficos simples hasta mapas de red e informes más elaborados. La información con la que se trabaja en el frontend no es más que los datos proporcionados por otra capa importante en la arquitectura de Zabbix, el "**backend**". En líneas generales, el backend es la capa que toma la información que los agentes han enviado al servidor (típicamente almacenada en una Base de Datos MySQL) para después suministrarla al frontend en el que se visualizarán los datos y enviará las correspondientes alertas que se desencadenen en función de la información recopilada.

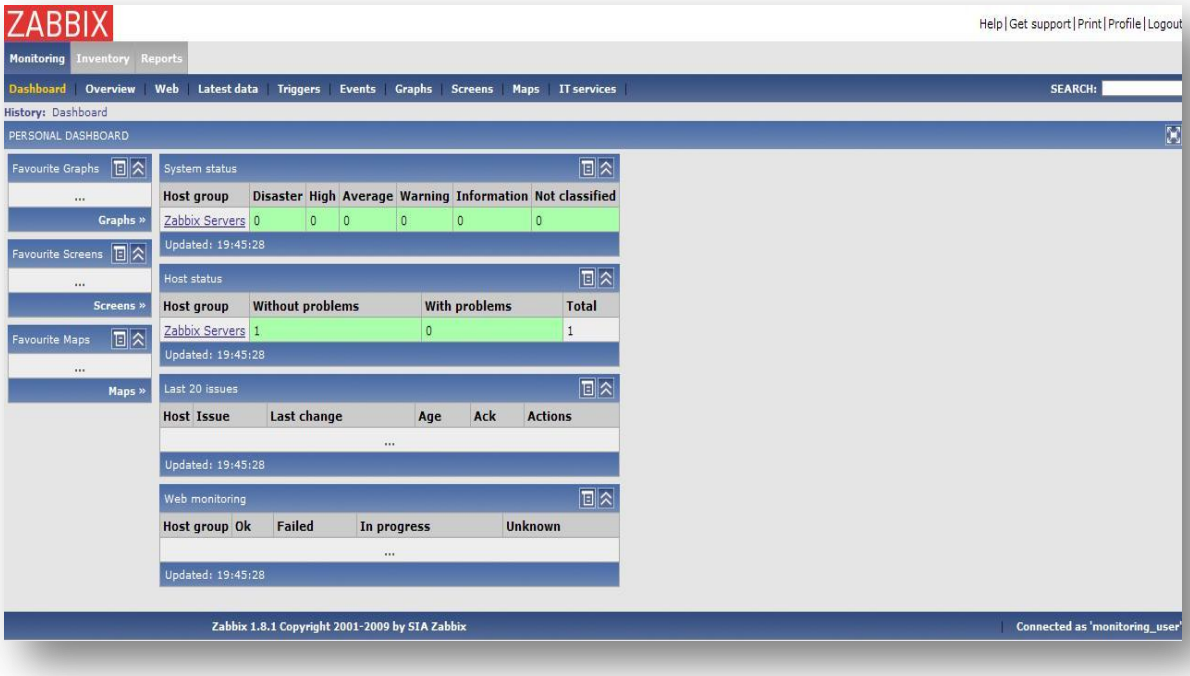



**Figura 3. Panel de control del frontend de Zabbix**

En la fase de despliegue profundizaremos en mayor detalle en las posibilidades que nos ofrece la interfaz web de la herramienta Zabbix.

## 3.2.4. Requisitos de Zabbix

Los requisitos exigidos por la solución de monitorización Zabbix se agrupan en dos categorías importantes: **requisitos hardware** y **requisitos software**.

### Requisitos Hardware

Los requisitos hardware dependen mayoritariamente de la configuración. Es difícil establecer unos requisitos específicos, por lo que cada instalación debería evaluarlos de forma individual. En cualquier caso, se establecen unas directrices mínimas a las que debería ajustarse un sistema en el cual se desee instalar Zabbix.

**Memoria física.**

Zabbix puede funcionar incluso con una cantidad de memoria RAM de 128MB, aunque, en general, se recomienda disponer de más memoria con objeto de que el sistema funcione con mayor rapidez.

**CPU.**

En general, la CPU utilizada no es un aspecto especialmente crítico, pues incluso un sistema basado en procesador Pentium II debería ser perfectamente capaz de ejecutar Zabbix. Sin embargo, las consideraciones en este caso son exactamente las mismas que para la memoria física, y es que cuanto más potente sea el procesador con el que trabajemos, mayores prestaciones obtendremos.

La Base de Datos sobre la cual trabaja Zabbix requiere de un consumo de CPU que se incrementa en función del tipo de Base de Datos escogido, así como del número de hosts monitorizados y de la cantidad de parámetros de éstos que son monitorizados.

La siguiente tabla nos sirve para hacernos idea de los requerimientos en cuanto a CPU y memoria que supone la instalación de Zabbix en función de los parámetros antes mencionados:

| Tamaño instalación | Plataforma | CPU/Memoria | Base de Datos | N° hosts monitorizados |
|---|---|---|---|---|
| Pequeña | Ubuntu Linux | PII 350MHz 256MB | MySQL MyISAM | 20 |
| Mediana | Ubuntu Linux 64 bit | AMD Athlon 3200+ 2GB | MySQL InnoDB | 500 |
| Grande | Ubuntu Linux 64 bit | Intel Dual Core 6400 | 4GB RAID10 MySQL InnoDB o PostgreSQL | >1000 |
| Muy grande | RedHat Enterprise | Intel Xeon 2xCPU 8GB | Fast RAID10 MySQL InnoDB o PostgreSQL | >10000 |

**Tabla 7. Ejemplos de configuración hardware**

**Espacio en disco duro.**

En cuanto al disco duro, el espacio necesario varía de forma directamente proporcional al tamaño de la Base de Datos y, como ya hemos visto, éste es un factor ligado al número de *hosts* que se desea monitorizar, a los datos a almacenar y al tiempo que se mantendrán éstos en la Base de Datos (histórico de valores).

Para hacernos una idea de los requerimientos en cuanto al espacio que ocupa la Base de Datos, supongamos que tenemos un total de 3000 valores monitorizados, valores que son refrescados cada 30 segundos. Esto supone que estaremos añadiendo datos a un ritmo de 3000/30 = **100** nuevos valores cada segundo. Tenemos la intención de mantener los valores dentro del histórico durante 30 días, de modo que, teniendo en cuenta el dato anterior de 100 valores por segundo, el resultado es de un número total de valores igual a 30 días * 24 horas/día * 3600 segundos/hora * **100** valores/segundo = **259.200.000**.

El tamaño que ocupa cada uno de esos valores depende del tipo de datos manejado (float, entero, string, fichero de log, etc.) aunque, de media, cada uno de esos valores obtenidos ocupa, físicamente, unos 50 bytes. Haciendo uso de número total de datos obtenido anteriormente, tenemos 259.200.000 * 50 bytes = **12 Gigabytes** de tamaño total para esos 30 días de histórico.

Sumado a lo anterior, Zabbix es capaz de elaborar estadísticas cada 30 minutos calculando máximos, mínimos, contadores por cada nuevo valor recogido y se encarga de almacenar en la Base de Datos en forma de *trends*. Los *trends* son básicamente datos que son utilizados para elaborar gráficos de tendencias en períodos de tiempo más amplios.

Habitualmente, y aunque varía del tipo de Base de Datos escogido, se necesitan de unos 128 *bytes* por cada total. Supongamos que queremos mantener los *trends* de datos durante 1 año. Si tenemos 3000 valores aproximadamente a monitorizar, éstos supondrán unos requerimientos de (3000/1800)*(24*3600*365)*128 = **6,3 Gigabytes** de tamaño.

Los eventos generados por Zabbix así como los envíos de alertas configurados también ocupan espacio dentro de la Base de Datos, de manera que cada uno de ellos supone unos 130 *bytes*. Tomando el peor de los casos, Zabbix generará un evento por segundo, lo cual significa que, en caso de desear mantener durante 1 año los eventos generados, necesitaríamos un espacio de 365*24*3600*130 = **3,82 Gigabytes**.

Por tanto, podemos calcular el espacio total necesario como el resultado de sumar el espacio que supone la instalación de Zabbix y las aplicaciones necesarias (incluyendo la Base de Datos), la conservación del histórico de datos, los datos de tendencias o *trends* y los eventos.

**Espacio total = Instalación y configuración + Histórico + Tendencias + Eventos** (1)

Considerando nuevamente un hipotético número de 3000 elementos a monitorizar, tendremos que, según la expresión anterior (1), el espacio total que necesitaremos es el siguiente:

- Instalación y configuración: habitualmente en torno a **10 Mbytes**.
- Histórico de valores: conservamos histórico cada 30 días, lo cual supone **12 Gigabytes**.
- Datos de tendencias: se conservan durante 1 año, dejando un total de **6,3 Gigabytes**.
- Eventos: los datos registrados sobre eventos son mantenidos igualmente durante 1 año, necesitando para ello **3,82 Gigabytes**.

Sumando estos datos obtenemos un total de **32,12 Gigabytes** de espacio necesario mínimo en el disco para mantener la configuración deseada para la instalación, el histórico de datos, las tendencias y los eventos.

Aunque, a priori, no deja de ser una necesidad de espacio respetable, se ha de tener en cuenta que dicho espacio no se ocupará de forma instantánea al instalar la plataforma, sino que irá evolucionando y creciendo progresivamente hasta alcanzar los totales considerados en las políticas anteriormente descritas.

## Requisitos Software

El sistema operativo del servidor central sobre el que se instala Zabbix puede ser uno de los siguientes:

- Linux.
- Solaris.
- HP-UX.
- AIX.
- FreeBSD.
- OpenBSD.
- OS X.
- SCO Open Server.

A nivel de software, los requisitos se enumeran en la siguiente lista:

- Apache versión 1.3.12 o posterior.
- PHP versión 5.0 o posterior. Igualmente son necesarios los módulos php-gd en versión 2.0 o posterior y php-bcmath.
- En cuanto a la Base de Datos, se escoge entre uno de estos posibles tipos:
    - MySQL versión 3.22 o posterior, siendo necesario también php-mysql.
    - Oracle versión 9.2.0.4 o posterior, incluyendo php-oci8.

- PostgreSQL, versión 7.0.2 o posterior. Será indispensable php-pgsql. Este tipo de Base de Datos se recomienda en caso de trabajar con un gran número de equipos y a un alto rendimiento.
- SQLite, versión 3.3.5 o posterior, necesitando asimismo php-sqlite3.

Si se desea hacer uso de la funcionalidad de envío de alertas sms se deberá instalar una utilidad que permita enviar dichos mensajes desde el servidor central.

La instalación y compilación de las fuentes de Zabbix implica tener que contar con los siguientes paquetes en función del tipo de Base de Datos:

- MySQL: paquete mysql-dev.
- PostgreSQL: paquete postgresql-dev.
- SQLite: paquete sqlite3-dev.

Sumado a todo lo anterior, se precisa de los siguientes elementos:

- *NET-SNMP* o *UCD-SNMP*, con sus correspondientes ficheros de cabecera y librerías, y así poder dar soporte al protocolo SNMP instalado en la plataforma y que posteriormente se discutirá. En la plataforma a implantar se instala *snmp* como aplicación NET-SNMP y *snmpd* como agente NET-SNMP, ambas en versión 5.2.3-4. De igual manera, son necesarios los paquetes libsnmp9, libsnmp9-dev y libsnmp-base.
- Librerías y ficheros de encabezamiento *Iksemel*, para así activar el servicio de mensajería instantánea Jabber.
- Librerías y ficheros de encabezamiento *libcurl*, versión 7.13.1 o posterior para activar la monitorización WEB.
- Compilador C, recomendado *GNU C*.
- *GNU Make*, con objeto de poder ejecutar los Makefiles de Zabbix.

Si se desea obtener información en mayor detalle sobre la herramienta Zabbix, existe un sitio web al efecto **[1]**.

## 3.3. SNMP

SNMP son las siglas de "*Simple Network Management Protocol*" o "*Protocolo Simple de Administración de Red*". Definido sobre la capa de aplicación (nivel 7 de la torre de protocolos OSI), SNMP es un protocolo que permite la labor de supervisión de la red, definiendo mecanismos de administración remota para equipos, en nuestro caso los diversos *hosts* monitorizados por la plataforma. Igualmente, permite analizar y comunicar información de estado entre dichos hosts, de forma que así se puedan

descubrir y analizar problemas en la infraestructura y proporcionar mensajes de estado de la misma. El objetivo principal de SNMP es crear una forma estandarizada de obtener información acerca de equipos independientemente del hardware subyacente.

El protocolo SNMP consiste en un grupo de estándares para la administración de redes que incluye un protocolo de capa de aplicación, un esquema de Base de Datos y un conjunto de objetos de datos. Prácticamente la totalidad de los fabricantes de hardware y de software ofrecen soporte a esos estándares, y los sistemas operativos más comúnmente utilizados pueden proporcionar información utilizando SNMP. Microsoft ofrece SNMP para sus plataformas Windows; todos los sistemas basados en UNIX disponen de demonios SNMP que reciben peticiones desde otros equipos.

La torre de comunicaciones SNMP está situada sobre la estructura de protocolos TCP/IP y, al estar basado en el protocolo UDP[4] **[2]**, necesita menos recursos que TCP. Además, UDP utiliza un solo paquete para enviar una solicitud o una respuesta.

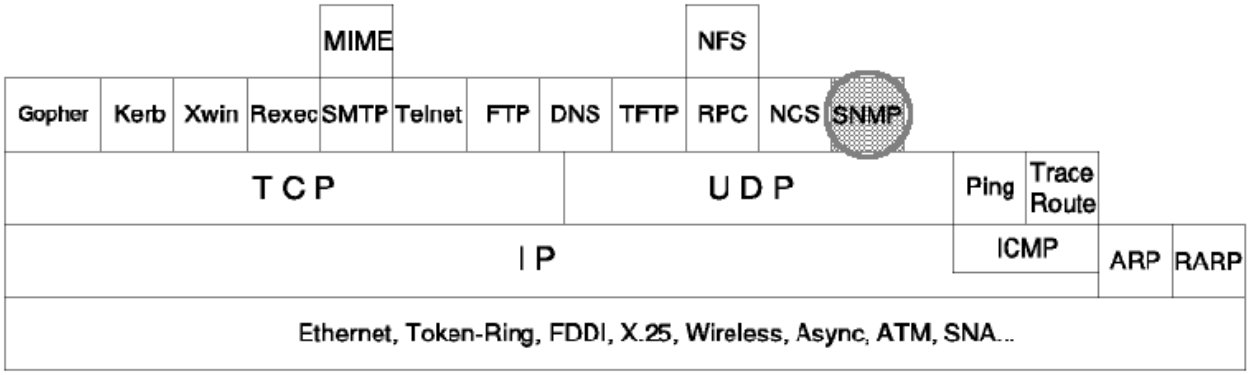



**Figura 4. Ubicación del protocolo SNMP**

Su principio de funcionamiento se basa en que la totalidad de administración de equipos se realiza mediante la manipulación de variables.

## 3.3.1. Utilidad de SNMP. Cómo funciona

La funcionalidad del protocolo SNMP sigue una arquitectura cliente-servidor y se construye a partir de un sistema cuyos componentes trabajan de manera conjunta. Estos componentes son los siguientes:

- **Base de gestión de información** (MIB, *Management Information Base*): la MIB es una colección de información sobre la red. Dicha información se almacena en una Base de Datos relacional de objetos gestionados a los que se puede

[4] UDP, User Datagram Protocol

acceder utilizando protocolos de gestión de red como SNMP. Cada uno de los objetos gestionados puede representar una característica de un determinado dispositivo a través de un cierto valor. Este valor puede hacer referencia a conceptos como la temperatura de un router, el estado de un disco duro, etc. La estructura de la MIB es tal que todos los objetos de gestión en SNMP se sitúan siguiendo una estructura arborescente.

- **Estructura de la gestión de la información** (SMI, *Structure of Management Information*): el SMI es un árbol jerárquico en el que se definen los tipos de datos que son permitidos en la MIB y cómo los objetos de la misma pueden ser nombrados y caracterizados. Típicamente los objetos de la MIB tienen seis atributos:

  a) Nombre. El nombre de un objeto se define como un *OBJECT IDENTIFIER* y se utiliza para nombrar los objetos gestionados. El nombre puede estar definido en tres tipos de MIBs.

     ✓ MIB estándar de Internet
     Mib OBJECT IDENTIFIER
     ::={ internet mgmt(2) 1 }

     ✓ MIB experimental
     Experimental OBJECT IDENTIFIER
     ::={ internet 3 }

     ✓ MIB privadas
     Mib OBJECT IDENTIFIER
     ::={ internet private(4) 1}

  b) Identificador: secuencia de enteros no negativos separados por puntos. Incluye el tipo de objeto, el nivel de acceso, restricciones de tamaño y la información del rango del objeto.

  c) Campo de sintaxis: la sintaxis del objeto establece el tipo de datos que lo modela. La sintaxis puede ser abstracta (utilizada para describir las estructuras de datos a intercambiar y la información de gestión contenida en las estructuras) o de transferencia (referida a las reglas de codificación básicas). En este campo se reflejan los tipos permitidos para los objetos (tipos simples, como integer, octet string, object identifier), los tipos de aplicación, los tipos estructurados (sequence, sequence of) y los subtipos (ipAddress, gauge, timeticks, etc.).

  d) Campo de acceso: define el nivel de acceso al objeto y puede tomar uno de cuatro valores posibles (*read-only*, *read-write*, *write-only*, *not-accessible*).

e) Campo de estado: define los requisitos de implementación de un objeto. Adopta uno de tres valores posibles (*Mandatory*, *Optional*, *Obsolete*).

f) Descripción textual: texto que describe el significado del objeto. Incluye el nombre completo del objeto y las versiones que identifican el tipo de dispositivo, sistema operativo y software de red.

- **Agentes SNMP**: todos los dispositivos de la red que van a ser monitorizados a través de SNMP necesitan tener incorporado un agente que ejecute los objetos de la MIB que sean relevantes. El agente proporciona la información contenida en la MIB a las aplicaciones de gestión cuando se le solicita.

### 3.3.2. Versiones de SNMP

El protocolo SNMP tiene varias versiones a través de las cuales un agente puede comunicarse:

- **SNMPv1 (versión 1)**: el modelo de seguridad de esta primera versión es poco sofisticado. Todos los dispositivos que se comunican sobre SNMPv1 utilizan el string de la comunidad para verificar si la solicitud se puede llevar a cabo. Por defecto, el string de la comunidad *private* permite tanto leer como escribir información, mientras que el string de la comunidad *public* sólo permite operaciones de lectura.
- **SNMPv2 (versión 2)**: introduce mejoras en términos de funcionamiento y de seguridad. Permite recuperar todas las entradas de una tabla en una sóla operación y soluciona los problemas de monitorización y de carga de tráfico de la versión 1. La implementación más común para la versión 2 es **SNMPv2c**[5], que utiliza las características de la versión 2 sin implementar el nuevo modelo de seguridad, sino utilizando el mecanismo de string de la comunidad introducido en la versión 1.
- **SNMPv3 (versión 3)**: la versión 1 utiliza como mecanismo de autenticación el parámetro referente a la comunidad, por lo que, si el agente y el administrador lo conocen, pueden interactuar entre ellos. Este tipo de protección es muy débil porque el texto se transmite en claro y puede explotarse mediante fuerza bruta. Para evitar esa falta de seguridad en las transmisiones, se creó una capa como complemento a las versiones 1 y 2, añadiendo a los mensajes SNMPv1 o SNMPv2 una cabecera adicional, dando lugar a lo que conocemos como la versión 3 del protocolo SNMP.

[5] SNMPv2c, Community-based Simple Network Management Protocol 2

## 3.4. Pandora FMS

Dentro del catálogo de soluciones de monitorización nos encontramos, además de a Zabbix, a Pandora FMS[6], que, al igual que Zabbix, es un software de código abierto cuyo fin es permitir la monitorización de un sistema dado y que involucra a un cierto número de elementos.

Las posibilidades de monitorización de un sistema que ofrece Pandora FMS abarcan desde parámetros simples como el espacio en disco, la carga del procesador, la memoria disponible hasta ideas más elaboradas como pueden ser el número de hilos de ejecución de un servidor Apache, el porcentaje de paquetes rechazado por el Firewall, el número de virus detectados por el sistema antivirus instalado, etc. Para la recolección de ciertos parámetros será necesario contar con un agente instalado en el equipo a monitorizar.

Otro aspecto común a Zabbix es la posibilidad de no sólo monitorizar, sino de también informar de anomalías en el sistema y de que los datos puedan ser guardados para su posterior análisis a lo largo del tiempo, detectando así tendencias (llamados *trends* en Zabbix) que permitan adelantarnos a los problemas antes de que estos se produzcan.

Enumeramos a continuación las principales características de Pandora FMS:

- Monitorización de sistemas de forma remota mediante TCP/IP o de forma local a través de agentes. Los agentes son abiertos y se pueden adaptar para la monitorización de cualquier elemento.
- Almacenamiento de los datos recogidos en la monitorización en una Base de Datos relacional.
- Elaboración de gráficos combinando datos de varios tipos y generación de informes SLA[7].
- Configuración de alertas para envío de mensajes vía e-mail o SMS.
- Gestión centralizada de las incidencias que ocurran en los sistemas.
- Disponibilidad de una consola de recepción de *traps SNMP*[8] en tiempo real.
- Soporte multiusuario con diferentes perfiles.
- Monitorización de hasta 10.000 parámetros diferentes, lo cual supone aproximadamente unos 1200 sistemas.

Se pueden encontrar más características sobre Pandora FMS en su site oficial **[29]**.

[6] Flexible Monitoring System
[7] Service Level Agreement (Acuerdo Nivel Servicio)
[8] El agente SNMP notifica a la consola de administración acerca de los cambios por medio de una interrupción.

### 3.4.1. Requisitos Hardware Mínimos

Se presentan aquí los requisitos necesarios para la instalación de un sistema de monitorización basado en Pandora FMS.

#### Requisitos para la consola y el servidor Pandora

Definiremos los requisitos en función del número de equipos (*agentes*) y del número de distintos grupos de elementos a monitorizar (*módulos*).

| N° agentes | N° módulos | Memoria | CPU |
|---|---|---|---|
| 500 | 5.000 | 2 GB | Un solo núcleo a 2GHz. |
| 2.000 | 10.000 | 4 GB | Doble núcleo a 2.5 GHz. |
| >4.000 | Indefinido | 12 GB | Cuatro núcleos a 3 GHz. |

**Tabla 8. Requisitos hardware para la consola y el servidor Pandora**

### 3.4.2. Requisitos software mínimos

El agente de Pandora puede ser ejecutado en cualquier hardware que tenga instalado uno de los siguientes sistemas operativos:

- Windows 2003 Server.
- Windows XP.
- Windows Vista.
- Windows 7.
- Windows 2008 Server.
- SUSE Linux 10.
- Ubuntu Linux 8.04.
- Debian Linux.
- AIX 4.3.3.
- HP-UX 11.x.
- Solaris 2.6.

### 3.4.3. Requisitos para el servidor Pandora

Aunque puede trabajar sobre cualquier sistema operativo que disponga de *Perl 5.8* con *iThreads* habilitados, el servidor Pandora está soportado únicamente sobre sistemas basados en UNIX, siendo SUSE y cualquier distribución basada en Debian las recomendadas.

De igual forma, Pandora necesita de un servidor MySQL para almacenar la información que van recogiendo los agentes. Dicho servidor puede instalarse sobre cualquier plataforma soportada por el propio MySQL, recomendándose plataformas Windows por su mayor eficiencia y rendimiento.

Se deberá tener instalado *Perl 5.8* para que el servidor funcione correctamente, además de los paquetes de *SNMP* (*net-snmp*) correspondientes al sistema operativo para así poder utilizar el servicio *snmp* de Pandora. En cuanto al resto de paquetes, se deberá disponer del paquete *nmap* y, opcionalmente, del paquete *xprobe2* para utilizar las características avanzadas de *reconserver*[9] así como las bibliotecas *traceroute* de Perl para poder llevar a cabo los autodescubrimientos de red. Igualmente es necesario el cliente binario de *WMI*[10] para hacer consultas contra sistemas Windows.

### 3.4.4. Requisitos para la consola

Al igual que en el caso del servidor Pandora, para la consola se recomienda su ejecución sobre sistemas basados en UNIX. Aunque la interfaz está construida puramente sobre *Apache*, *MySQL*, y *PHP*, podría trabajar teóricamente sobre cualquier sistema que soporte este tipo de software.

### 3.4.5. Requisitos para administrar la herramienta vía WEB

Se deberá disponer de un navegador web para instalar y comprobar el funcionamiento de la consola. En principio no se recomienda tener instalado *Flash* como requisito indispensable, aunque es necesario si se desea hacer uso de las gráficas interactivas presentadas bajo dicho formato.

### 3.4.6. Dependencias de paquetes

Pandora FMS depende no sólo del sistema operativo en lo que a configuración software se refiere. Además, necesita una serie de paquetes adicionales que no siempre vienen instalados por defecto. En la fase de implantación del proyecto se mencionarán los paquetes necesarios para una instalación completa de Pandora FMS.

[9] Servidor de reconocimiento que "descubre" equipos conectados a la infraestructura de red facilitando así su integración en la monitorización de Pandora.
[10] Windows Management Instrumentation, Instrumentos para la gestión de Windows.

## 3.5. Nagios

Siguiendo la línea de la misma familia de sistemas en la que encuadramos a Zabbix y Pandora FMS, encontramos a Nagios, aplicación igualmente concebida para la monitorización de sistemas y de redes. Nagios ofrece posibilidades que abarcan desde la monitorización de los nodos conectados a la red hasta servicios especificados por el usuario y, al igual que los otros sistemas estudiados, su razón de ser se basa en el desarrollo de una estrategia proactiva ante determinados problemas.

Del mismo modo, se disponen de herramientas de notificación con las que alertar al usuario en caso de se haya producido cualquier tipo de problema en el sistema o red monitorizados. El comportamiento y configuración de estas notificaciones es muy similar al caso de Zabbix, teniendo la posibilidad de personalizarlas bajo determinados parámetros definidos por el usuario.

Nagios fue, en sus inicios, desarrollado para funcionar bajo sistemas operativos Linux, aunque, en principio, puede funcionar instalado en otros sistemas operativos basados en UNIX. Internamente está escrito en C y, tal como Zabbix y Pandora, Nagios es **Software Libre** (Licencia GPL versión 2).

En cuanto a las características específicas de Nagios, señalamos las siguientes:

- Monitorización de servicios de red: SMTP, POP3, HTTP, SSH, DNS, etc.
- Funcionalidad *soft states/hard states* dirigida a evitar falsas alarmas en los datos de monitorización (ver ANEXO XIII. Soft States/Hard States en Nagios).
- Monitorización de recursos: carga de procesador, espacio libre en sistemas de ficheros, uso de memoria, etc.
- Capacidad de desarrollar plugins de forma sencilla y flexible para permitir a los usuarios definir sus propias comprobaciones.
- Posibilidad de definir una topología o jerarquía de red que permita distinguir los servicios caídos o inaccesibles.
- Envío de notificaciones vía email, SMS, mensajería instantánea, etc.
- Definición de eventos cuya ejecución esté asociada a un determinado problema, facilitando así el camino en la búsqueda de la estrategia proactiva mencionada anteriormente.
- Agrupación de los elementos monitorizados, permitiendo así el envío de avisos y alertas según el grupo al que el elemento pertenezca.
- Entorno web integrado para la visualización del estado actual de los servicios, generación de estadísticas, historial de alarmas, etc.
- Asignación de roles a los usuarios, de manera que puedan dividirse las tareas a ejecutar según el papel desempeñado.
- Soporte a bases de datos para el almacenamiento de datos externo.
- Ampliación de funciones mediante la instalación de *plugins* o complementos.

### 3.5.1. Requisitos de sistema

El único requisito para poder ejecutar Nagios es disponer de un equipo con un sistema operativo Linux (o basado en UNIX) instalado y un compilador del lenguaje de programación C. Será igualmente necesario tener configurada correctamente la red para permitir la correcta ejecución de las comprobaciones efectuadas por Nagios.

No es absolutamente obligatorio utilizar los *CGIs*[11] incluidos con Nagios, pero, en caso de hacerlo, se precisará tener instalado el siguiente software:

1. Servidor web (recomendado ***Apache***).
2. Librería ***gd*** de Thomas Boutell con versión 1.6.3 o posterior (necesaria para los mapas de estado y los CGIs de datos de tendencias o *trends*) **[3]**.

El desarrollo de Nagios se debe fundamentalmente a Ethan Galstad y, al igual que todo el software libre, a mucha gente anónima que contribuye notificando fallos en el programa, proporcionando soluciones a los mismos, creando plugins, etc. Se pueden encontrar más detalles en el fichero "LICENSE" o en la licencia *GNU GPL* publicada en Internet **[4]**.

Actualmente Nagios cuenta con más de 250.000 usuarios en todo el mundo y algunas de las entidades más destacables que hacen uso de este software son la Junta de Andalucía, la Consejería de Educación, etc.

Para más información sobre Nagios, visitar la dirección **[5]**.

[11] CGI, Common Gateway Interface

# 4

# Análisis del sistema de videovigilancia

# 4. ANÁLISIS DEL SISTEMA DE VIDEOVIGILANCIA

Esta etapa de análisis marca los problemas a abordar en el proyecto, constituyendo una guía para todo el desarrollo del mismo además de una referencia de acuerdo entre la parte que plantea la necesidad del proyecto y la parte que finalmente la desarrolla.

Así pues, en este punto se determinan las necesidades reales que quedarán cubiertas una vez se haya estudiado el entorno del sistema de videovigilancia.

El resultado obtenido de ello será una definición de requisitos en la que se especifique claramente "***qué***" se persigue con la idea de construir una plataforma de monitorización de los equipos que conforman el sistema CCTV de la UC3M.

Como punto de partida comenzaremos con una visión de la configuración actual de la infraestructura tecnológica de este especial entorno que es el sistema de videovigilancia.

## 4.1. Infraestructura tecnológica del sistema de videovigilancia

Se mostrará en este punto el conjunto de equipos que forman la infraestructura en la que se basa el sistema CCTV de la UC3M, incluyendo la topología de red existente y las funciones de los equipos que se desean monitorizar en la plataforma.

### 4.1.1. Topología de la red

La característica más reseñable de la infraestructura de red del sistema que nos ocupa es que se trata de una **red privada**, aislada de la red de área local existente en la UC3M con objeto de preservarla así de intrusiones de terceros que pudieran comprometer la privacidad y confidencialidad de la información registrada por el sistema de videovigilancia.

Simbólicamente, el siguiente gráfico explica la concepción de la red de videovigilancia desde la perspectiva de la privacidad:

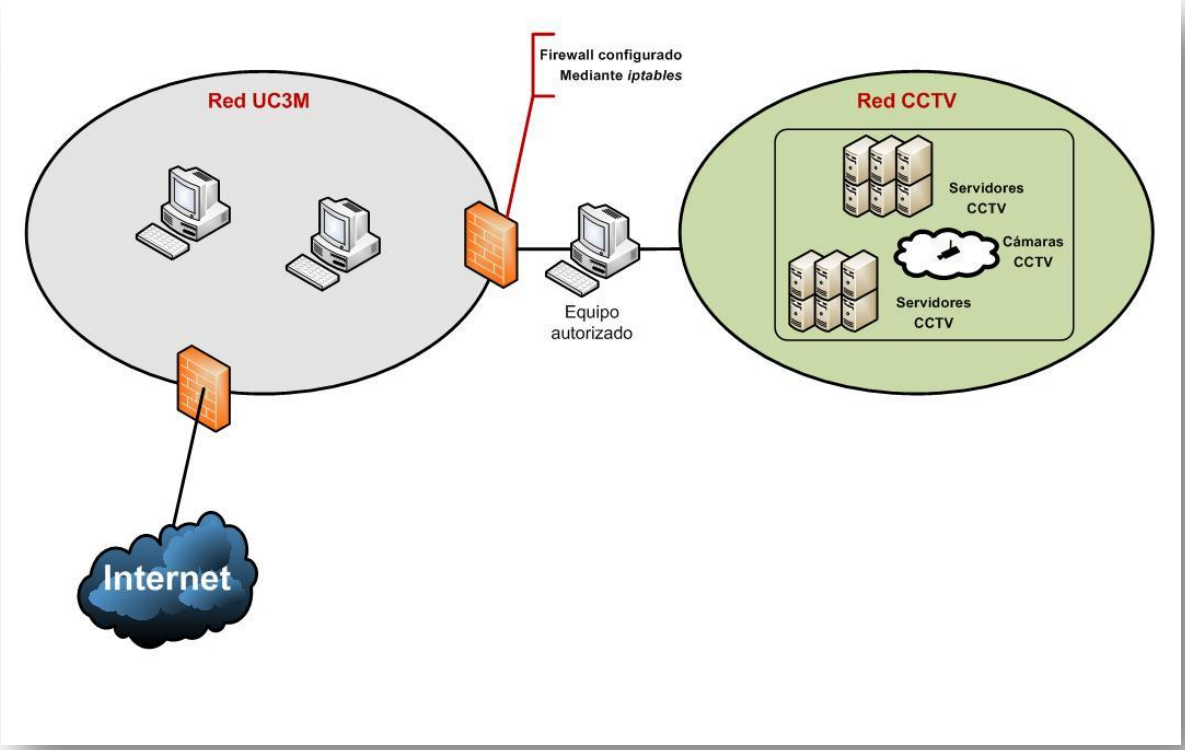



**Figura 5. Esquema de la infraestructura de red**

Actualmente, dentro de la infraestructura de videovigilancia se dispone de 12 servidores de grabación (6 en el campus de Leganés y 6 en el campus de Getafe), 246 cámaras de vídeo (121 en el campus de Leganés y 125 en el campus de Getafe) y 4 equipos alojados en los centros de control (2 en el campus de Leganés y 2 en el campus de Getafe), que son ubicaciones específicas dentro de la UC3M desde las cuales el personal de seguridad física se encarga de monitorizar en tiempo real las imágenes recogidas por las cámaras instaladas en los campus. Estos últimos equipos, si bien no son críticos desde el punto de vista de la disponibilidad, son igualmente monitorizados en la plataforma ya que, al ser equipos constantemente conectados a los servidores de grabación, son de suma utilidad a la hora de detectar fallos en estos últimos.

Tan importante como los servidores de grabación y las cámaras de vídeo lo es el equipamiento de red gracias al cual todas esas entidades quedan interconectadas entre sí. Por ello, se decide incluir en la monitorización los distintos sistemas que conforman la infraestructura de la red de videovigilancia, como los conmutadores de planta a los cuales se conectan las cámaras en cada uno de los edificios de la UC3M, los conmutadores de la red CCTV a los que se conectan los servidores de grabación así como los equipos de interconexionado que permiten la comunicación entre el sistema de videovigilancia del campus de Leganés y el sistema análogo en el campus de Getafe.

El esquema que sigue la implementación actual del sistema de videovigilancia en ambos campus en términos de topología es el siguiente:

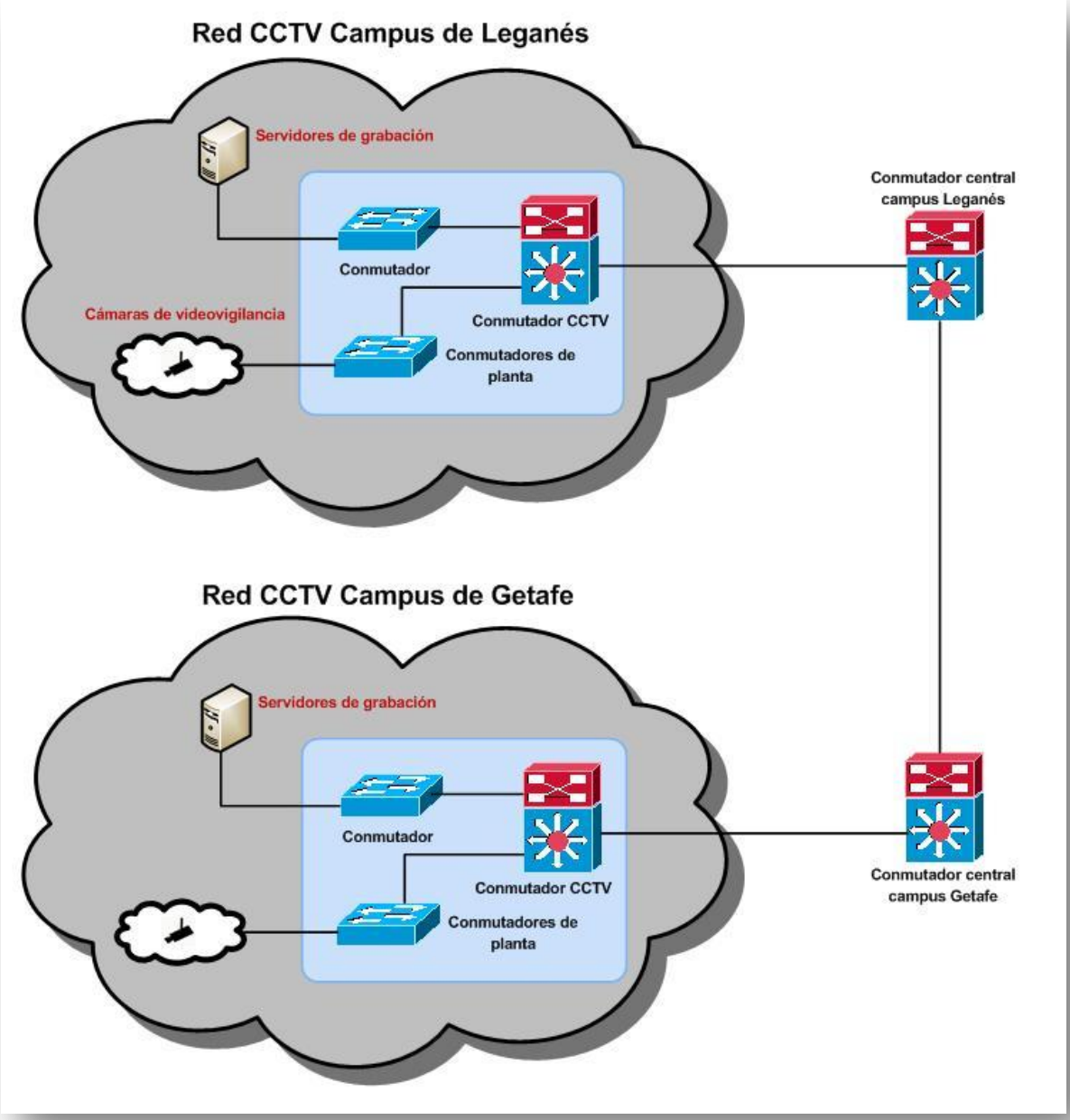



**Figura 6. Topología de red del sistema de videovigilancia**

Las funciones de cada elemento concreto de red se describen posteriormente en base a la función de cada uno de ellos (conmutador central de CCTV, conmutador central de campus, conmutadores de planta).

A nivel físico, los equipos implicados en la topología de red del sistema de videovigilancia se observan en la figura que sigue (ver siguiente página):

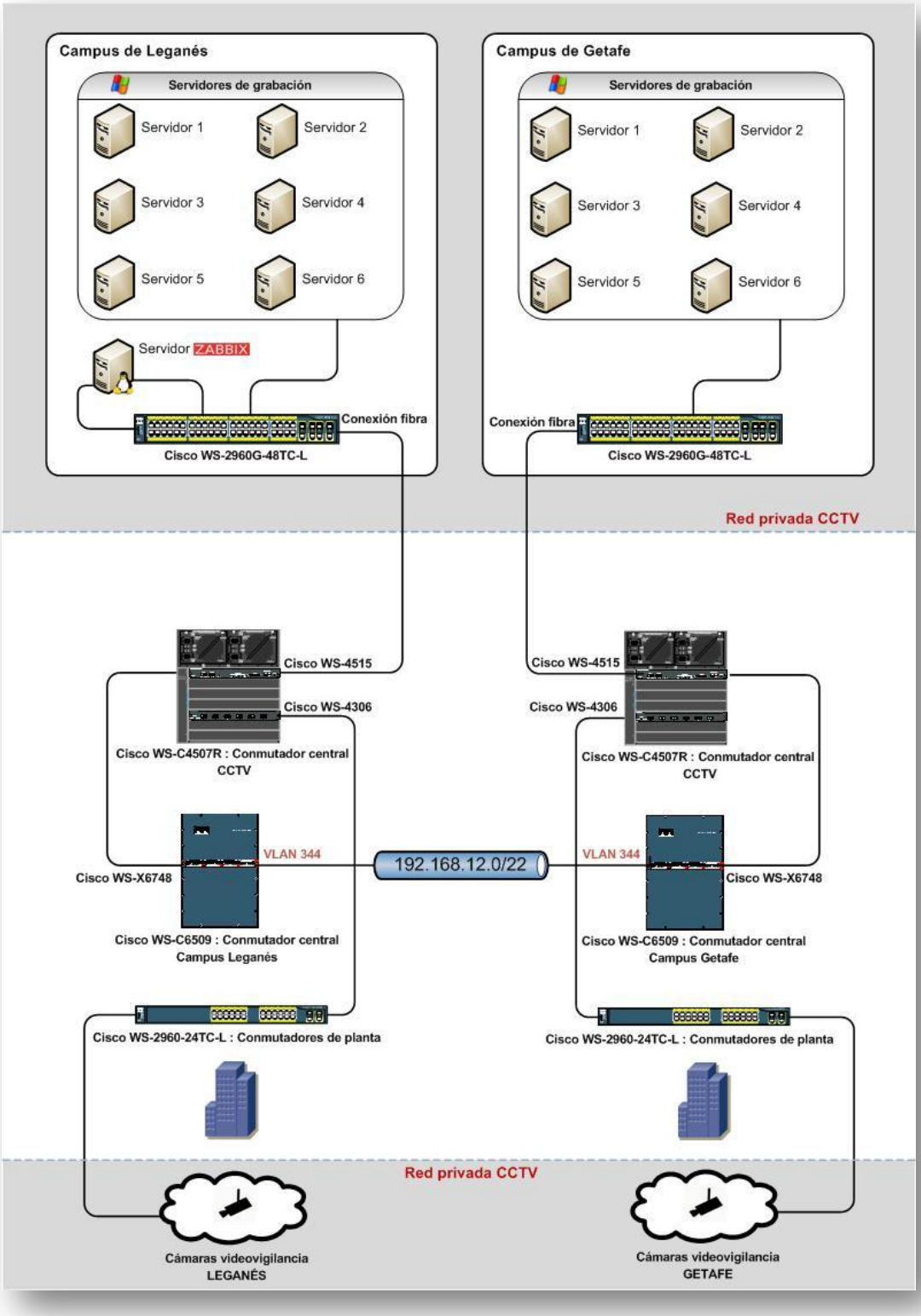



**Figura 7. Topología de red del sistema de videovigilancia a nivel físico**

Dentro de la red privada de CCTV, la subred con la que contamos en el entorno de monitorización es la siguiente:

- Red de los servidores de grabación, cámaras de videovigilancia y centros de control en los campus de Getafe y Leganés: **192.168.12.0/22.**

Tal como puede observarse en la **Figura 7**, el servidor que alojará la plataforma Zabbix de monitorización se encuentra en el campus de Leganés y está provisto de dos interfaces. La primera de ellas está conectada a la red privada de CCTV (VLAN 344) y la segunda está configurada sobre el segmento 131 de la red de la UC3M (VLAN 2). De esta manera, el servidor de Zabbix jugaría el papel de equipo autorizado tal como se indicó en la **Figura 5**, pudiendo comunicarse tanto con los servidores de grabación y cámaras de videovigilancia como con ciertos equipos integrados en la red de la UC3M. Las restricciones de acceso que indican qué equipos externos a la red privada de CCTV pueden establecer conexión con Zabbix se implementan vía *iptables*[12] **[41]** definiéndose las políticas que ayuden a garantizar en todo momento la seguridad del servidor Zabbix y de la red privada CCTV. Tales políticas están configuradas de tal manera que se aceptan conexiones sólo desde equipos de la red de CCTV y desde ciertos equipos de la red de la UC3M.

Además, la capacidad de enrutamiento entre las dos interfaces se deshabilita desactivando la opción de *forwarding*.

Cada uno de los servidores se conecta a su respectivo conmutador ubicado dentro de los armarios en los que éstos se encuentran instalados y, a su vez, cada uno de esos conmutadores se comunica con su correspondiente conmutador central de campus de CCTV. Finalmente, la comunicación entre los campus de Getafe y de Leganés corre a cargo de los conmutadores centrales de campus y la conexión a la red de las cámaras de videovigilancia queda establecida a través de los conmutadores de planta asignados a los edificios en los que dichas cámaras se encuentran instaladas, siguiendo las normas de cableado estructurado.

El estándar de conectividad seguido en términos de transmisión de datos es de ***Gigabit Ethernet*** para las conexiones de los servidores de grabación y los equipos de control con sus respectivos equipamientos de red, y ***Fast Ethernet*** para el enlace de las cámaras de videovigilancia con los correspondientes conmutadores de planta.

[12] Sistema de firewall vinculado al kernel de Linux

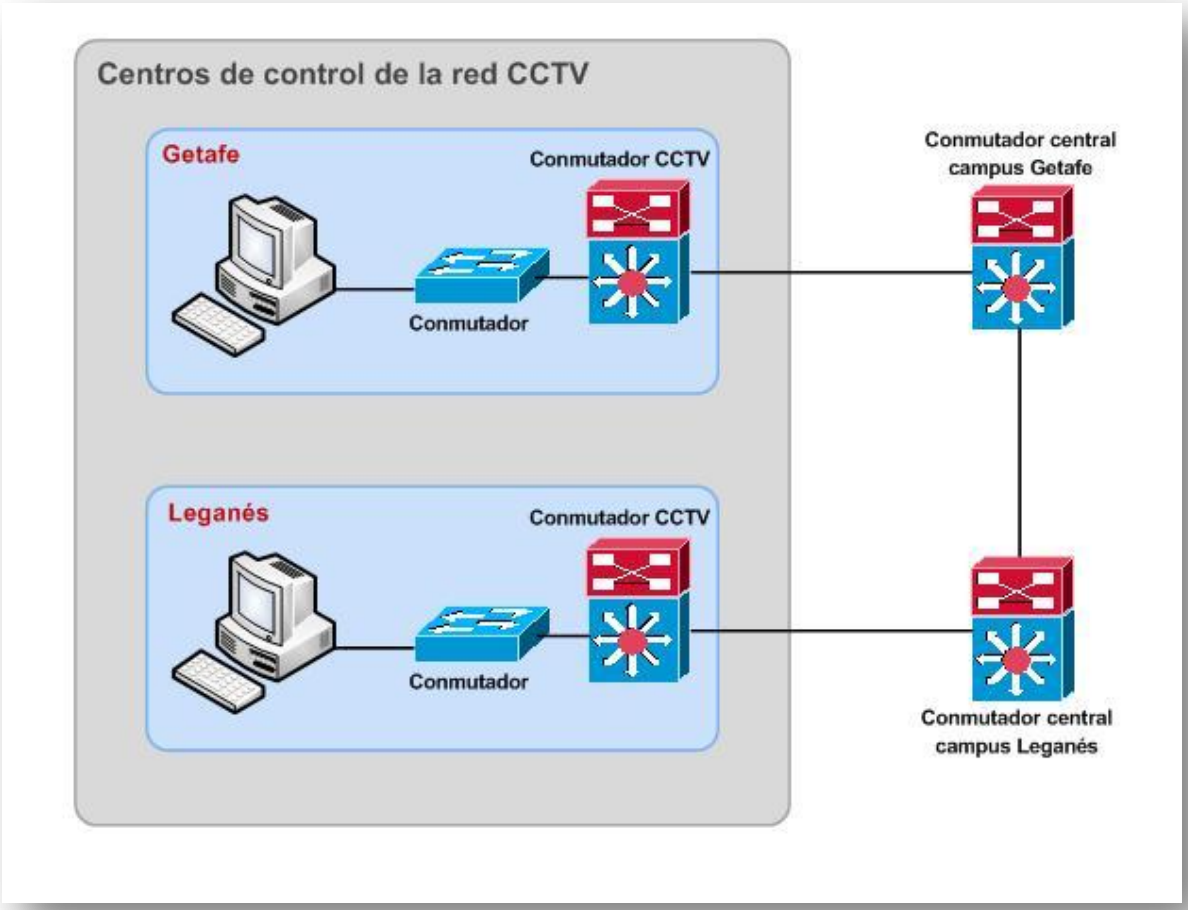



**Figura 8. Topología de red para los equipos de los centros de control**

## 4.1.2. Equipos que forman la red de videovigilancia

Diferenciaremos los siguientes grupos dentro de la red del sistema de CCTV:

- Servidores de grabación
- Cámaras de videovigilancia
- Equipos de los centros de control
- Conmutadores centrales de Campus en la red CCTV
- Conmutadores centrales de Campus en la red UC3M
- Conmutadores de planta en los campus

## Servidores de grabación

Hay un total de 12 servidores de grabación, de los cuales 6 se encuentran en el campus de Leganés y los 6 restantes, en el campus de Getafe.

Estos servidores suponen el pilar de la infraestructura de videovigilancia ya que de su correcto funcionamiento depende la disponibilidad del sistema de CCTV. Si uno de estos servidores falla, las cámaras conectadas a él no podrán monitorizarse y, peor aún, no podrá crearse ninguna grabación de las imágenes recogidas por ellas.

Es por ello que el rendimiento de estos servidores así como de las aplicaciones en ellos instaladas y directamente relacionadas con la videovigilancia debe ser controlado para actuar de manera conveniente ante cualquier problema que pudiera presentarse.

Identificaremos esos 12 servidores según el campus al que pertenecen con los siguientes rangos de red:

- **Servidor 1**
- **Servidor 2**
- **Servidor 3**
- **Servidor 4**
- **Servidor 5**
- **Servidor 6**

→ **Rango 192.168.14.0/24** → **Campus de Leganés**

- **Servidor 1**
- **Servidor 2**
- **Servidor 3**
- **Servidor 4**
- **Servidor 5**
- **Servidor 6**

→ **Rango 192.168.12.0/24** → **Campus de Getafe**

Las especificaciones hardware de los servidores son exactamente las mismas tanto en el campus de Getafe como en el campus de Leganés, a excepción de las capacidades de los discos duros en ellos instalados.

En el campus de Leganés los servidores de grabación están configurados sobre una ***caja Supermicro SC833T550B de montaje en rack, 3u, con fuente de 550w y 8 bahías para discos duros SATA***. La placa base, modelo ***Supermicro x3000-PD i3010 con soporte para RAID 5 y hasta 8 GB de RAM,*** también es común a todos los servidores del campus de Leganés, e igualmente están provistos de un procesador ***Intel DualCore Pentium E2180 a 2Ghz, 2 GB de memoria RAM*** y ***Controladora SATA II 3ware 9550sx-8LP*** como elementos hardware más importantes.

En el caso de los servidores del campus de Getafe, las especificaciones son exactamente las mismas que en el campus de Leganés, a excepción del modelo de procesador, tratándose en este caso del modelo ***Intel DualCore Pentium E2160 a 1,8 Ghz.***

Uno de los aspectos más críticos a nivel de hardware en los servidores de grabación lo constituye el sistema de almacenamiento. Éste se encuentra implementado en los 12 servidores en la forma de un sistema ***RAID***[13] **de nivel 5** [27] gestionado por la controladora 3ware mencionada anteriormente y compuesto de **8 discos SATA II** cuyas capacidades, según cada servidor, son las siguientes:

- Para los servidores 1, 2, 3 y 5 del campus de Leganés: **8 discos SATA II de 250 GB de capacidad**.
- Para todos los servidores del campus de Getafe: **8 discos SATA II de 320 GB de capacidad**.
- Para los servidores 4 y 6 del campus de Leganés: **8 discos SATA II de 500 GB de capacidad**.

[13] Redundant Array of Independent Disks

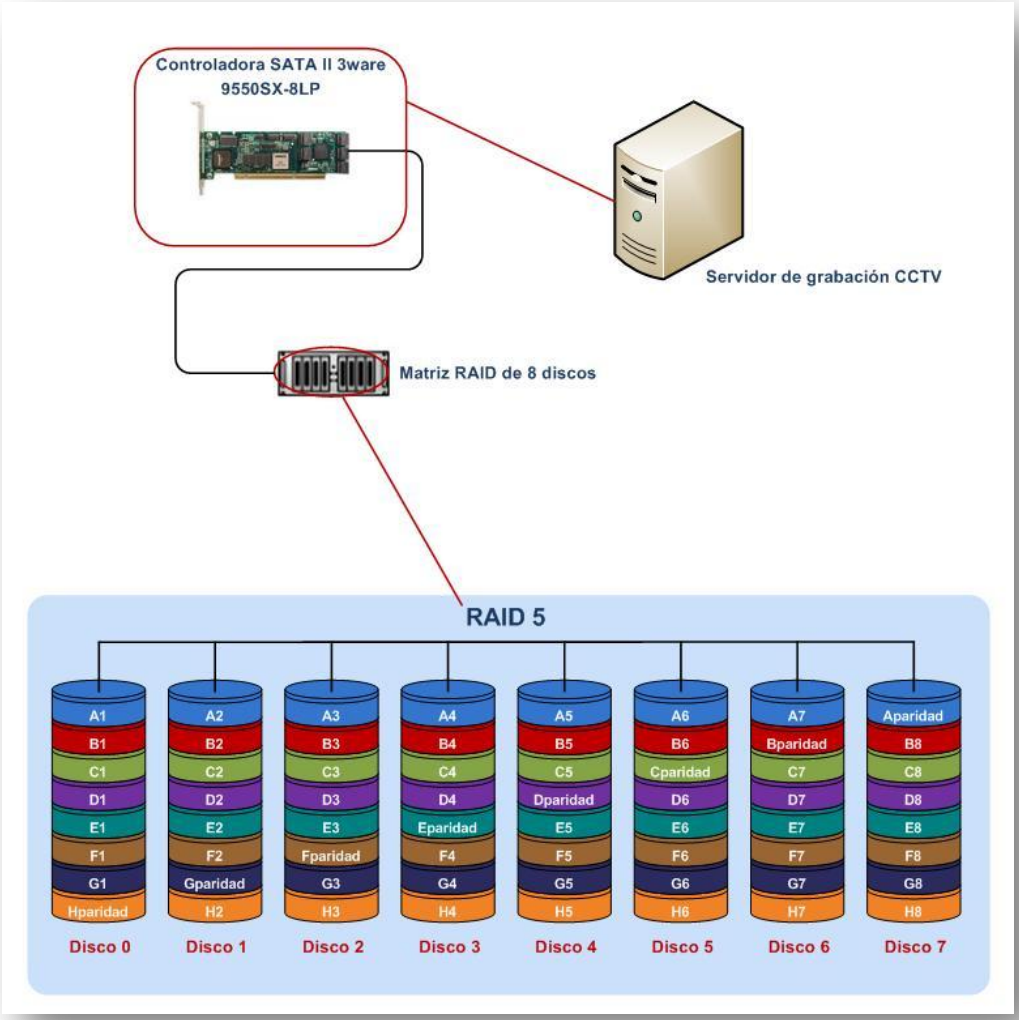



**Figura 9. Esquema del Sistema RAID de almacenamiento en los servidores de grabación**

Los sistemas RAID de almacenamiento suponen un método de combinación de dos o más discos de manera que todos ellos formen una única unidad lógica en la cual se almacenarán de forma redundante los datos. Con esta tecnología se protegen los datos contra los fallos de disco duro y, en caso de producirse éstos, los servidores de grabación seguirían estando activos y en funcionamiento hasta que se sustituyese el disco en el que se generó el error. Existen distintos niveles y en todos ellos el denominador común es la pérdida de parte de la capacidad de almacenamiento en favor de los datos de paridad y la redundancia.

En el caso de la implementación escogida para los servidores de grabación de CCTV, **RAID 5**, los datos y la paridad son guardados en los discos que componen el propio RAID, con lo que se conseguirá aumentar la velocidad de demanda al satisfacer cada disco sus peticiones independientemente del resto de unidades. Así, se obtienen unas

elevadas prestaciones, una muy alta protección de datos y se soportan múltiples accesos de lectura y escritura simultáneas, todo ello aprovechándose hasta el 80% de capacidad de los discos.

Sobre la implementación del RAID 5, en uno de los servidores se encuentra implementada una variante en la que 2 de esos 8 discos funcionan como "***Hot Spare***" (los 6 restantes configuran un RAID 5 clásico). El mecanismo *Hot Spare* añade aún más fiabilidad al sistema de almacenamiento en caso de fallo, pues, en caso de que uno de los 6 restantes discos presente un error en su funcionamiento, uno de esos 2 discos activos como *Hot Spare* pasará automática e inmediatamente a sustituir a esa unidad averiada.

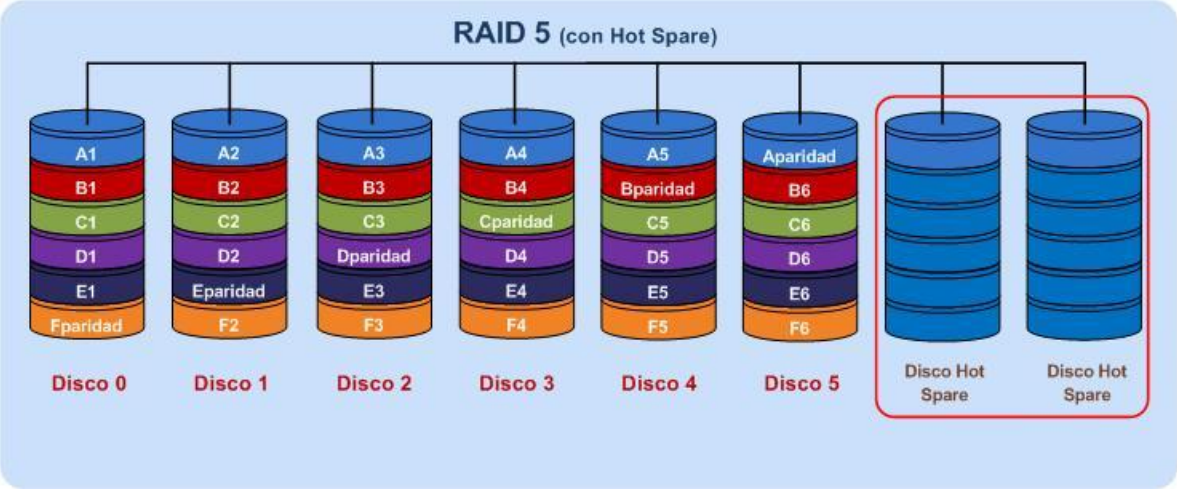



**Figura 10. RAID 5 con Hot Spare**

En cuanto al **software** de los servidores, todos ellos cuentan con **Windows XP Professional con Service Pack 2** como sistema operativo instalado.

El resto de software instalado en los servidores de grabación se detalla en la siguiente lista:

- ***Sony RealShot Manager, versión 4.3.1.12***: esta aplicación propietaria de Sony es esencial para la monitorización y grabación de imágenes recogidas por las cámaras de videovigilancia. Dicha aplicación, de la que puede encontrarse más información en la referencia **[6]**, será convenientemente monitorizada en la plataforma ya que de su correcto funcionamiento depende la disponibilidad de las imágenes obtenidas por las cámaras de videovigilancia ya sea en tiempo real o a través de grabaciones correspondientes a otros instantes en el tiempo. Para más información sobre este software, ver ANEXO VII. Software de gestión Sony RealShot Manager™.

- **3ware 3DM2 web interface**: interfaz gráfica de usuario basada en web a través de la cual se posibilita la monitorización de la tarjeta controladora de discos SATA II instalada en cada servidor. Las prestaciones ofrecidas incluyen la evaluación y gestión del sistema RAID así como de las unidades de disco independientes que lo constituyen.

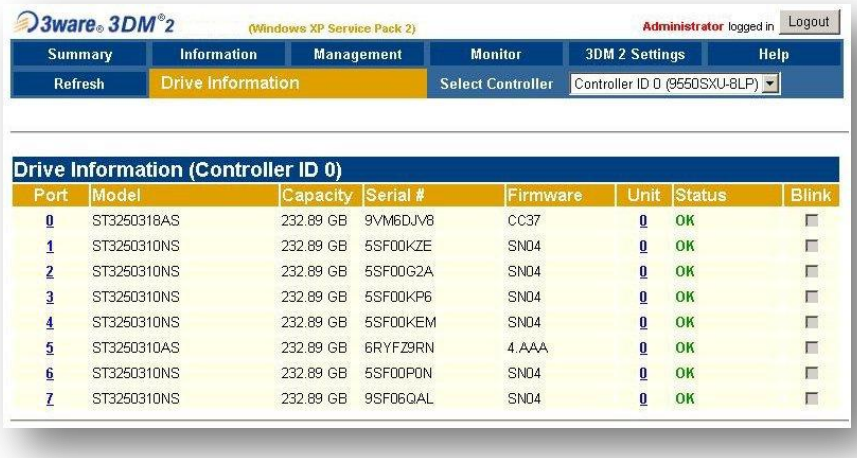



**Figura 11. Monitorización del Sistema RAID a través de la interfaz web 3DM2**

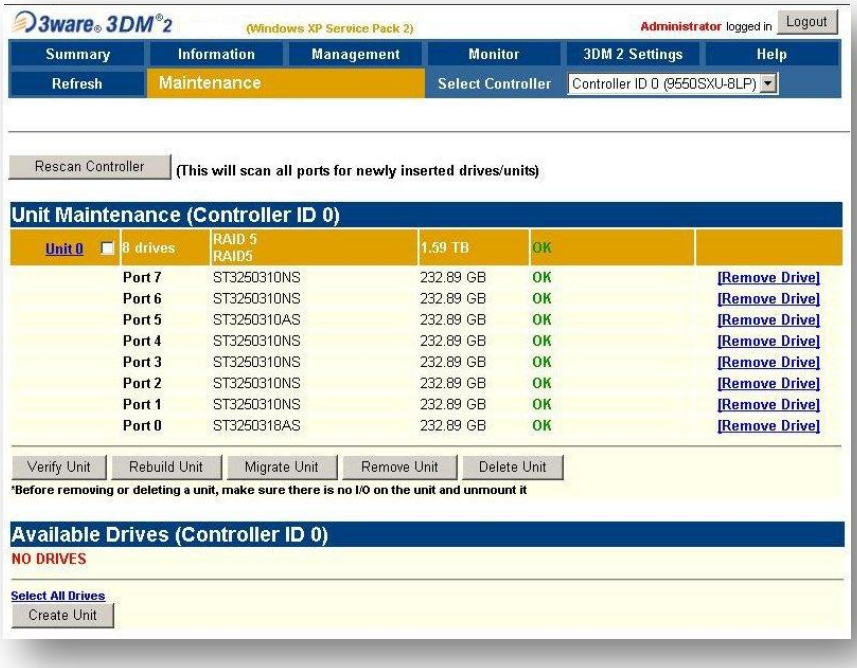



**Figura 12. Monitorización de discos a través de la interfaz web 3DM2**

- **3ware SNMP Agent**: a través del agente SNMP del fabricante 3ware se pueden obtener datos adicionales a los recogidos por la propia interfaz web de 3DM2, como el estado y temperatura de cada disco o la capacidad y estado del sistema RAID.
- **Supermicro Supero Doctor Client**: este software está concebido para monitorizar el estado de los servidores en términos de refrigeración y temperatura. Así, instalando su agente SNMP correspondiente, permite conocer el estado y velocidad de giro de cada unos de los 6 ventiladores instalados en cada servidor así como la temperatura del sistema y de la CPU y los distintos niveles de voltaje. Para más información, visitar [7].


**Figura 13. Monitorización del sistema de refrigeración y temperaturas vía Supero Doctor**

- **Servicio SNMP Windows**: para poder recoger datos de los agentes SNMP provistos por los distintos fabricantes es condición indispensable tener instalado el servicio SNMP del propio sistema operativo Windows.
- **Intel PRO Network Connections SNMP Agent**: la monitorización del tráfico de red en cada servidor es otro de los aspectos fundamentales para su seguimiento, razón por la cual se instala este agente para captar, en cada interfaz de red, parámetros como los paquetes y bytes recibidos o los paquetes y bytes enviados.
- **Agente Zabbix**: el agente Zabbix es el software necesario para que cada servidor pueda enviar los datos de monitorización al servidor en el que quedará instalada la plataforma de monitorización final.
- **Agente PandoraFMS**: para hacer las pruebas comparativas con Pandora FMS como sistema de monitorización se debe tener igualmente instalado un agente capaz de comunicarse con el servidor de Pandora.

- **Agente Nagios**: las consideraciones en este caso son las mismas que para los sistemas de Zabbix y Pandora, siendo necesario disponer del agente correspondiente a Nagios para el envío de datos al servidor de Nagios que se instalará para evaluar las diferencias y similitudes entre éste y Zabbix.
- **PostgreSQL admin 8.2**: la versión del software de *Sony RealShot Manager* mencionado anteriormente que se encuentra instalada en los servidores de grabación es la revisión **4.3.1.12** y, para el almacenamiento de los ficheros de vídeo, hace uso de una BBDD PostgreSQL cuya gestión y mantenimiento se realiza a través del software "*PostgreSQL admin*". Para más información, está disponible el sitio web **[8]**.

## Cámaras de videovigilancia

Las cámaras de videovigilancia que forman parte del circuito cerrado de televisión están desarrolladas, al igual que el software de grabación, por el fabricante Sony.

Los modelos de cámaras utilizados en los campus de la UC3M son los siguientes:

- **SNC-CS50P [23]**: cámara fija (sin capacidad de movimiento). Cuenta con función "*Día/Noche*" para captar imágenes tanto en entornos diurnos como nocturnos y ofrece capacidad de procesado de vídeo con detección de movimiento, entradas de sensor y salidas de alarma, almacenamiento en una tarjeta de memoria *ATA* de imágenes previas y posteriores a la alarma y acceso simultáneo para que hasta un total de 20 usuarios o clientes puedan acceder concurrentemente a la cámara y así puedan evaluar las imágenes recogidas por la misma.
- **SNC-DF80P [24]**: cámara minidomo fija. Las características que presenta se resumen en el análisis de vídeo inteligente, formato de compresión seleccionables (*JPEG, MPEG-4, H.264*), función de "*Día/Noche*" análoga al modelo CS50P, mecanismo de montura de objetivo "*Ball-Joint*" **[9]** patentado por Sony y que permite rotar la óptica de la cámara en cualquier dirección, y, finalmente, cuenta con detección inteligente de movimiento y de objetos con los que minimizar falsas alarmas.
- **SNC-RX550P [25]**: cámara de toma panorámica móvil. Este modelo concreto de cámara es igualmente móvil y dispone de un mecanismo de movimiento que permite 360 grados de giro horizontal ininterrumpido con inclinación y zoom 26x. Sus especificaciones son similares a las de los modelos de cámaras mencionados hasta ahora y, sobre el modelo DF80P, añade capacidad de codificación dual para las imágenes, de forma que puede crear éstas en formato *MPEG4* y JPEG de manera simultánea.

En resumen, y sumado a todo lo descrito anteriormente, las principales características comunes a todos los modelos de cámara utilizados son:

- Compresión MJPEG[14]/MPEG4 conmutable.
- Funcionalidad *ARC* que, de cara a posibles inestabilidades de la red, ajusta el refresco de imagen a las condiciones de la red mediante cálculos efectuados en base al *RTT*[15].
- Entradas de alarma por contacto y salida de relé para activación de dispositivos asociados, como un foco de infrarrojos, por ejemplo.
- Función de "*detección de actividad*", que convierte a la cámara en un sensor de movimiento que dispara la alarma correspondiente en el software de grabación ante la presencia de objetos o personas que se muevan.
- Función *Multicasting*, para permitir la difusión eficiente de la imagen recogida por la cámara en redes cuyo hardware admita este tipo de transporte.
- Almacenamiento de imágenes en pre-alarma y post-alarma.
- Envío automatizado de imágenes (por lapso de tiempo o por alarma) vía correo electrónico a través de servidores SMTP y/o servidores FTP.
- Seguridad en red garantizada a través de las listas de usuarios con diferentes niveles de privilegios y por medio de filtrado IP.
- Función "*Día/Noche*" y análisis avanzado de movimiento y detección de objeto abandonado.

Las especificaciones técnicas de cada cámara se pueden encontrar en tablas en el ANEXO VIII. Especificaciones de las cámaras de videovigilancia.

De cara a la monitorización, las especificaciones técnicas de relevancia que ofrecen las cámaras son sus **prestaciones de red**. El denominador común a todas ellas es que soportan los protocolos TCP/IP, ARP, ICMP, HTTP, FTP (cliente/servidor), SMTP, DHCP, DNS, NTP, RTP/RTCP, SNMP (MIB-2). En esta lista nos centraremos en 3 protocolos importantes: ICMP, http y SNMP (MIB-2). A través de ICMP y de HTTP se obtendrá el estado de la cámara en términos de conectividad y, por medio de SNMP, se recogerán datos como el tráfico de red actual de la cámara o la versión del firmware instalada en cada una de ellas.

El concepto de funcionamiento de una cámara y su conexión con los servidores de grabación queda descrito en la siguiente figura:

[14] Motion JPEG
[15] Round Trip Time

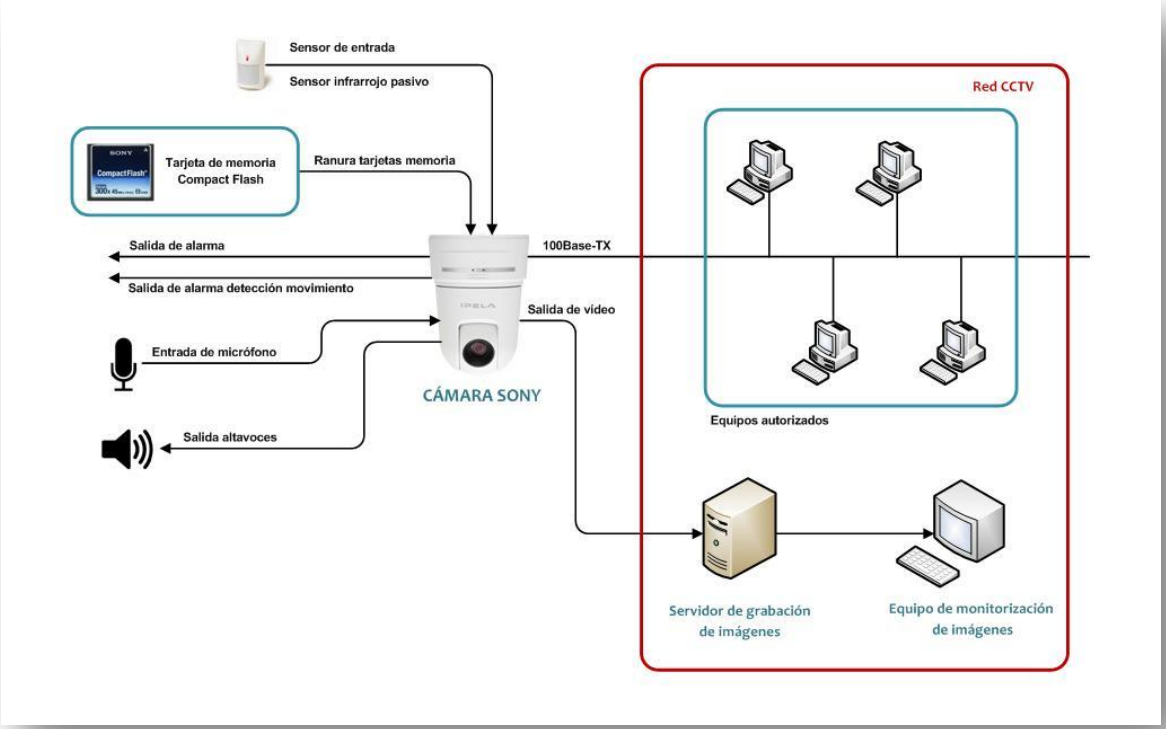



**Figura 14. Configuración de funcionamiento de una cámara Sony**

En el sistema implementado actualmente en la UC3M, el acceso a las cámaras de videovigilancia está restringido a los equipos pertenecientes a la red privada del sistema de videovigilancia. Desde estos equipos autorizados únicamente se pueden modificar ciertos parámetros de configuración de las cámaras o monitorizar la imagen que estén captando. Por su parte, los servidores de grabación pueden acceder directamente tanto a las imágenes recogidas en el momento actual por las cámaras como a grabaciones de momentos anteriores almacenadas en sus discos duros. Los equipos de monitorización de los centros de control son clientes conectados a los servidores de grabación, por lo que también cuentan con los mismos privilegios para el visualizado de las imágenes obtenidas por las cámaras en tiempo real así como de las grabaciones almacenadas. Esa dependencia entre los equipos de los centros de control y los servidores de grabación hace que la disponibilidad de estos últimos cobre aún mayor importancia.

Al margen de los servidores de grabación y los equipos de monitorización ubicados en los centros de control de cada campus, el resto de equipos integrados en la red privada de CCTV de la UC3M puede conectarse a las cámaras de vídeo a través de la interfaz web de la que éstas disponen y desde la cual se pueden efectuar cambios en su configuración así como examinar las imágenes que en ese mismo momento esté captando la cámara. Para acceder desde el explorador Web a los menús de

monitorización y administración de la cámara bastará con introducir la dirección IP de ésta en la barra del explorador. Sin embargo, si se desea ajustar los parámetros de la cámara a través del menú correspondiente, se deberá iniciar sesión con el usuario (y contraseña) autorizado a ello. Para más información sobre las diferentes posibilidades de configuración que ofrece una cámara Sony, ver ANEXO X. Configuración de las cámaras de videovigilancia Sony.

## Equipos de los centros de control

Desde los puestos de control central se monitorizan las cámaras del sistema de videovigilancia en tiempo real a través de una serie de equipos y pantallas instaladas al efecto. Todos estos equipos forman parte de la red privada de videovigilancia por lo que el acceso a los mismos se puede realizar únicamente desde otros equipos conectados a la misma red.

Las características hardware más relevantes para los equipos del centro de control de Leganés son:

- Placa base Supermicro Dual Core XEON 15000X
- 2GB de memoria RAM
- 2xCPU Intel XEON a 2,33 Ghz QUAD CORE con tecnología de 45nm, 1333 Mhz Bus.
- Caja Supermicro torre 865W silenciosa con 8 SAS/SATA HOT SWAP.
- 8 discos duros de 250 GB SATA-II con velocidad de hasta 7200 rpm, 8 MB de caché, del fabricante *SEAGATE*.
- Tarjeta gráfica PNY Nvidia Quadro 4 440 NVS 256 MB DDR3 PCI-Express X16, 2 conectores LHF con soporte para cuatro monitores.
- 2xHD SEAGATE 73 GB SAS 15000 rpm 16MB 3,5 pulgadas CHEETA 15K.5
- Regrabadora DVD LG H54N doble capa dual 2MB 16XDVDR / 8XDVDRW / 48XCDR.
- Disquetera 31/2 1.4MB.

En el aspecto software, los equipos de los centros de control de ambos campus tienen, como sistema operativo, ***Windows XP Professional con SP2*** y cuentan con las siguientes aplicaciones instaladas:

- ***Sony RealShot Manager, versión 4.3.1.12***: al igual que en el caso de los servidores de grabación, se instala este software para la monitorización de las imágenes captadas por las cámaras y para la reproducción de grabaciones almacenadas. La diferencia radica en que el modo de instalación es tal que las

operaciones que se lleven a cabo con este software en los equipos del centro de control dependen de su instalación previa en los servidores de grabación.

- ***Agente Zabbix:*** del mismo modo que para los servidores de grabación, se instala este software necesario para el envío de los datos de monitorización a la plataforma que se implementará**.**
- ***Servicio SNMP***: para obtener datos de los equipos de los centros de control vía SNMP se necesita tener instalado este servicio del sistema operativo.

Existen otras aplicaciones, como el explorador Web necesario para iniciar sesión en las cámaras (*Internet Explorer*), pero que carecen de importancia desde el enfoque de monitorización que nos ocupa.

## Conmutadores centrales de campus en la red CCTV

Observando de nuevo la *Figura 6* (Topología de red del sistema de videovigilancia), vemos que las dos subredes existentes (**192.168.14.0/24** y **192.168.12.0/24**) se encuentran conectadas a sus respectivos conmutadores centrales de CCTV (subred **192.168.14.0/24** para el campus de Leganés y subred **192.168.12.0/24** para el campus de Getafe).

Su misión no es otra que interconectar, dentro de la subred de cada campus, el conjunto de servidores de grabación, cámaras de videovigilancia y equipos de monitorización del centro de control.

Los servidores de grabación se encuentran instalados en un armario rack desde el cual existe una conexión con el respectivo conmutador central. En los racks de los servidores se instala un conmutador intermedio que se encarga de conectar los servidores de grabación con los correspondientes conmutadores centrales de la red CCTV. Éstos se encargan de conectar la red CCTV con los conmutadores centrales de la red UC3M a través de los cuales será posible comunicar las dos subredes CCTV mencionadas anteriormente.

El modelo de conmutador para esos equipos intermedios es un ***Cisco Catalyst 2960G-48TC-L***. Se trata de un conmutador Ethernet inteligente que cuenta con un total de 48 puertos, de manera que a ellos pueden conectarse distintos dispositivos a los que se les provee de conectividad tanto *Fast Ethernet* como *Gigabit Ethernet* y de alimentación sobre el propio estándar *Ethernet (Power on Ethernet)*.

### Conmutadores centrales de campus en la red UC3M

La función cumplida por estos conmutadores es hacer "visibles" entre ellas las instalaciones de CCTV de los campus de Getafe y de Leganés. Así, de esta manera es posible acceder a los servidores de grabación del campus de Getafe desde equipos de la red CCTV de Leganés y viceversa.

### Conmutadores de planta en los campus

Siguiendo las normas de cableado estructurado, los conmutadores de planta se instalan en los edificios de cada campus para conectar los dispositivos (fundamentalmente cámaras de videovigilancia) con los conmutadores centrales de CCTV.

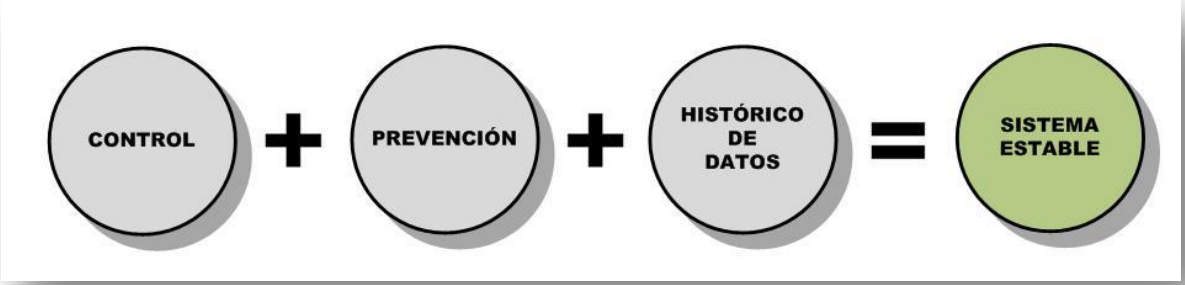
## 4.1.3. Necesidades generales

Después de todo lo expuesto hasta ahora, vemos claro que el funcionamiento del sistema de CCTV implementado en la UC3M en sus campus de Leganés y de Getafe depende de una infraestructura en la que es crítica la disponibilidad de los equipos informáticos correspondientes a los servidores de grabación. Es por ello por lo que surge la necesidad real de mantener constantemente operativos estos equipos, para lo cual deberá llevarse a cabo una tarea de seguimiento y supervisión a un nivel que sólo puede facilitarnos una herramienta de monitorización.

Tan importante como asegurar la disponibilidad, ya no sólo de los servidores de grabación, sino de todo el sistema en general, es poder prever futuros problemas y adelantarnos a éstos antes de que se produzcan. Para ello se hace necesario un análisis exhaustivo de todos los componentes y variables implicados en el entorno del sistema de videovigilancia que nos ocupa.

Además de todo lo anterior, existe una necesidad en términos de mantenimiento de una información histórica que permita observar el comportamiento de los distintos equipos y así, en un futuro, se posibilite llevar a cabo una labor de estudio en la que se identifiquen los puntos débiles del sistema así como soluciones que permitan eliminarlos o paliarlos en la medida de los posible.

La estrategia a seguir es adoptar una actitud proactiva en la que, frente a una postura reactiva en la que se reacciona una vez el problema ya se ha producido, se tomen decisiones antes de que surjan inconvenientes que pudieran comprometer el funcionamiento del sistema.




Estas tres ideas se traducen, en conjunto, en las siguientes necesidades:

- Control desde una misma ubicación de todas y cada una de las funciones más importantes de los equipos que forman parte del sistema. Lo ideal es poder disponer de una interfaz desde la que se pueda observar el comportamiento y rendimiento de los equipos en tiempo real.
- Programación de alertas mediante el envío de mensajes a través de medios utilizados frecuentemente, como el correo electrónico o un cliente de mensajería instantánea.
- Gestión de los equipos de forma que éstos se puedan añadir o eliminar, y que permita configurar los parámetros susceptibles de ser monitorizados.

Las necesidades aquí descritas surgen de sucesivas reuniones con los responsables del sistema de videovigilancia en las que se estima conveniente plantear una alternativa que mejorase el proceso de evaluación seguido hasta el momento.

## 4.1.4. Necesidades para cada tipo de equipo

Ya hemos visto que la infraestructura del sistema de videovigilancia está compuesta por equipos de diferentes categorías y con diferentes funciones. Por tanto, no monitorizaremos los mismos elementos en un servidor de grabación que en un conmutador de red, por ejemplo. Dividiremos los parámetros a monitorizar en función de los distintos tipos de equipo que hemos tratado en el punto **4.1.2**.

### Servidores de grabación

Los servidores de grabación tienen a Windows como sistema operativo, en su versión ***XP Professional***.

En cualquier caso, los parámetros comunes a monitorizar en la totalidad de los servidores, con independencia de la versión del sistema operativo Windows instalada, son los siguientes:

- **Información general:**
  - **Información del host:**
    - Nombre del equipo.
    - Sistema operativo.
    - Fabricante del procesador.
    - Estado del equipo (operativo o no).
    - Tiempo que lleva arrancado.

- **Disponibilidad:**
  - Almacenamiento:
    - Espacio libre en la unidad C: (en bytes).
    - Espacio libre en la unidad C: (en porcentaje sobre el total disponible).
    - Espacio total en la unidad C: (en bytes).
    - Capacidad del sistema RAID implementado (en bytes).
    - Estado de la unidad RAID (*correcto, verificación, inicialización, degradado, reconstrucción, recuperación, migración, inoperable, desconocido*).
    - Modelo de cada uno de los 8 discos que componen la unidad RAID.
    - Temperatura de cada uno de esos 8 discos duros (en grados centígrados).
    - Tiempo que lleva cada disco en funcionamiento (en días).
    - Estado de cada disco duro (*offline, OK*).
    - Número de discos conectados a la controladora RAID.
    - Porcentaje completado en la unidad RAID (este porcentaje se reserva para los estados del RAID *inicialización, reconstrucción* y *verificación*).

- **Memoria física y memoria *swap*:**
    - Memoria física libre.
    - Memoria *swap* libre.
    - Memoria física total.
    - Memoria *swap* total.

- **Rendimiento de la CPU:**
  - Carga del procesador.
  - Media de carga del procesador en los últimos 5 minutos.
  - Media de carga del procesador en los últimos 15 minutos.
  - Porcentaje de utilización del procesador.

- **Integridad de ficheros:**
  - Comprobación del *checksum* del fichero autoexec.bat
  - Comprobación del *checksum* del fichero config.sys

- **Procesos:**
  - Número de procesos que corren en el sistema.

- **Servicios y aplicaciones:**
  - Número de procesos *RealShot Manager* ejecutándose en el servidor (cuando este valor sea igual a cero, sabremos que el software de gestión de cámaras no está en ese momento disponible).
  - Estado del servicio SNMP de Windows (detenido o en marcha). De la disponibilidad de este servicio depende que los agentes SNMP de 3ware e Intel puedan obtener datos de los servidores.
  - Estado del servicio DHCP (por seguridad, se tiene deshabilitado en todos los servidores y se monitoriza su estado para verificar que no está iniciado).
  - Estado del servicio correspondiente al agente Zabbix (detenido o arrancado). Si éste no está iniciado, será imposible la toma de datos a través del servidor Zabbix.
  - Versión del agente Zabbix instalado.

- **Red:**
  - Conexiones TCP activas y establecidas.
  - Bytes recibidos por segundo en las interfaces de red (tráfico de entrada).
  - Bytes transmitidos por segundo (tráfico de salida).
  - Ping TCP al servidor.
  - Ping TCP al agente Zabbix.
  - Estado del servidor HTTP.

- **Windows Logs (visor de eventos de Windows):**
  - Log de *Aplicación* (registro con los eventos registrados por las aplicaciones o programas).
  - Log de *Sistema* (registro con los sucesos de seguridad, tales como intentos de inicio de sesión válidos y no válidos).
  - Log de *Seguridad* (registro que contiene los sucesos registrados por los componentes de Windows).

- **Refrigeración:**
  - Temperatura de la CPU.
  - Temperatura del sistema.
  - Velocidad (en rpm) de cada uno de los 6 ventiladores instalados en cada servidor.

## Cámaras de videovigilancia

Los datos que se desean extraer de las cámaras son fundamentalmente de información general, como el modelo, aunque también interesa obtener el tráfico de red registrado en cada una de ellas:

- **Información general:**
  - Modelo de cámara.
  - Versión del firmware instalado.
  - Número de serie.

- **Red:**
  - Respuesta a ping.
  - Estado del servidor HTTP.
  - Tráfico de red de entrada.
  - Tráfico de red de salida.

## Equipos de los centros de control

En estos equipos interesa monitorizar su tráfico de red, la aplicación *RealShot Manager* y otros parámetros acerca del sistema.

- **Información general:**
  - **Información del host:**
    - Nombre del equipo.
    - Sistema operativo.
    - Fabricante del procesador.
    - Estado del equipo (operativo o no).
    - Tiempo que lleva arrancado.

- **Disponibilidad:**
  - Almacenamiento:
    - Espacio libre en la unidad C: (en bytes).
    - Espacio libre en la unidad C: (en porcentaje sobre el total disponible).
    - Espacio total en la unidad C: (en bytes).

- **Memoria física y memoria *swap*:**
    - Memoria física libre.
    - Memoria *swap* libre.
    - Memoria física total.
    - Memoria *swap* total.

- **Rendimiento de la CPU:**
  - Carga del procesador.
  - Media de carga del procesador en los últimos 5 minutos.
  - Media de carga del procesador en los últimos 15 minutos.
  - Porcentaje de utilización del procesador.

- **Integridad de ficheros:**
  - Comprobación del *checksum* del fichero autoexec.bat
  - Comprobación del *checksum* del fichero config.sys

- **Procesos:**
  - Número de procesos ejecutándose en el sistema.

- **Servicios y aplicaciones:**
  - Número de procesos *RealShot Manager* ejecutándose en el servidor (cuando este valor sea igual a cero, sabremos que el software de gestión de cámaras no está en ese momento disponible).
  - Estado del servicio DHCP (por seguridad, se tiene deshabilitado en todos los servidores y se monitoriza su estado para verificar que no se encuentra iniciado).
  - Estado del servicio SNMP de Windows (detenido o en marcha).
  - Estado del servicio correspondiente al agente Zabbix (detenido o arrancado). Si éste no está iniciado, será imposible la toma de datos a través del servidor Zabbix.
  - Versión del agente Zabbix instalado.

- **Red:**
  - Conexiones TCP activas y establecidas.
  - Bytes recibidos por segundo en las interfaces de red (tráfico de entrada).
  - Bytes transmitidos por segundo (tráfico de salida).
  - Ping TCP al servidor.
  - Ping TCP al agente Zabbix.
  - Estado del servidor HTTP.

- **Windows Logs (visor de eventos de Windows):**
  - Log de *Aplicación* (registro con los eventos registrados por las aplicaciones o programas).
  - Log de *Sistema* (registro con los sucesos de seguridad, tales como intentos de inicio de sesión válidos y no válidos).
  - Log de *Seguridad* (registro que contiene los sucesos registrados por los componentes de Windows).

## Conmutadores centrales de campus en la red CCTV

Para toda la electrónica de red la información monitorizada será fundamentalmente acerca de sus conexiones físicas.

- **Información general:**
  - Ubicación del conmutador.
  - Modelo del conmutador.
  - Nombre del conmutador.
  - Tiempo que lleva en funcionamiento el conmutador.

- **Red:**
  - **Información sobre cada puerto:**
    - Estado administrativo de la interfaz.
    - Estado operativo de la interfaz.
    - Nombre alias para la interfaz.
    - Descriptor de la interfaz.
    - Bytes de entrada a la interfaz.
    - Bytes de salida desde la interfaz.

## Conmutadores centrales de campus en la red UC3M

- **Información general:**
  - Tiempo que lleva en funcionamiento el conmutador.
  - Carga de CPU.

- **Red:**
  - **Información sobre cada puerto:**
    - Bytes de entrada a la interfaz.
    - Bytes de salida desde la interfaz.

## Conmutadores de planta en los campus

- **Información general:**
  - Modelo del conmutador.
  - Nombre del conmutador.
  - Ubicación del conmutador
  - Tiempo que lleva encendido.

- **Red:**
  - **Información sobre cada puerto:**
    - Estado administrativo de la interfaz.
    - Estado operativo de la interfaz.
    - Bytes de entrada a la interfaz.
    - Bytes de salida desde la interfaz.
    - Descriptor de la interfaz.

### 4.1.5. Especificación de requisitos software

El personal responsable del sistema de videovigilancia de la UC3M plantea la necesidad de seguir un modelo de supervisión del sistema en el que se satisfagan las necesidades generales descritas en el punto 4.1.4.

Tras sucesivas reuniones en el área de seguridad informática se acuerda desarrollar una plataforma de monitorización basada en alguna solución software ya existente. El resultado de esas reuniones iniciales será una especificación de los requisitos que deberá cumplir dicha plataforma.

Esa especificación de requisitos es un catálogo en el que se incluye una completa definición de las necesidades que deberá cubrir la plataforma de monitorización a desarrollar. Para definir esa lista de requisitos se pueden seguir una serie de estándares pertenecientes a la *Ingeniería de Requisitos* como los que explicaremos a continuación.

#### Métrica V3

***Métrica Versión 3*** es una metodología de planificación, desarrollo y mantenimiento de sistemas de información creada por el *Ministerio de Administraciones Públicas* **[10]**. Los objetivos perseguidos con esta metodología son los siguientes:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico de desarrollo.
- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando mayor importancia al análisis de requisitos.
- Mejorar los Sistemas de Información permitiendo una mayor capacidad de adaptación a los cambios y considerando la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción del software a lo largo del ciclo de vida del proyecto, teniendo

en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.

- Facilitar la operación, uso y mantenimiento de los productos software obtenidos.
- Asegurar que el proyecto software cumple los objetivos definidos en términos de calidad, coste y plazos asignados a su producción.

Según **Métrica V3**, los requisitos se pueden clasificar como sigue (no se trata de una lista exhaustiva):

| TIPO | DESCRIPCIÓN |
|---|---|
| **Funcional** | Definen las capacidades o funcionalidades. |
| **De rendimiento** | Establecen las limitaciones de funcionamiento del software (velocidad de operaciones, uso de memoria, etc.). |
| **De seguridad** | Definen los criterios de seguridad para el software (restricciones de acceso, confidencialidad, integridad, etc.). |
| **De implantación** | Indican las características de entrega y puesta en marcha del software. |
| **De disponibilidad del sistema** | Definen los niveles de disponibilidad del software (cómo puede usarse el sistema cuando está operativo, su capacidad mínima y media disponible, etc.). |

**Tabla 9. Clasificación de requisitos según Métrica V3.**

Cada uno de esos tipos de requisitos deberá tener un conjunto de atributos:

| ATRIBUTO | DESCRIPCIÓN |
|---|---|
| **Identificador** | Nombre del requisito, conciso y no ambiguo, que hace referencia a un único requisito del sistema. |
| **Autor** | Nombre del analista del sistema de información encargado de definir con el usuario y/o cliente un determinado requisito. |
| **Tipo de requisito** | Especifica la clasificación del requisito. |
| **Descripción** | Especificación detallada del requisito para evitar ambigüedades o falta de información. Podrá incluir menciones a los datos de entrada y/o salida implicados. |
| **Prioridad** | Indica el orden temporal de realización del requisito. Puede tomar uno de siguientes valores: *alta, media, baja*. |
| **Estado** | Especifica la situación de requisito durante el ciclo de vida del software. Puede tomar uno de los siguientes valores: *propuesto, aprobado, incorporado.* |

| | |
|---|---|
| **Fecha de creación** | Fecha en que se especificó el requisito por primera vez. |
| **Fecha de revisión** | Fecha en la que se modificó el requisito una vez creado. |
| **Otros atributos** | Necesidad, protocolos usados, etc. |

**Tabla 10. Definición de los atributos de un requisito según Métrica V3.**

## Agencia Espacial Europea (ESA[16])

Los estándares definidos por la Agencia Espacial Europea proponen un conjunto de actividades para abordar el ciclo de vida de un proyecto de software y, aunque están concebidos para su aplicación en las prácticas software de la propia Agencia Espacial Europea, pueden ser adaptados para la realidad de cualquier otra empresa específica.

El ciclo de vida marcado por esos estándares está compuesto de seis fases:

- **UR:** definición de requisitos de los usuarios.
- **SR:** definición de los requisitos del software.
- **AD:** definición del diseño arquitectónico.
- **DD:** diseño detallado y producción de código.
- **TR:** transferencia del software a operaciones.
- **OM:** operación y mantenimiento.

La especificación de requisitos, al igual que en Métrica V3, constituye uno de los productos a lo largo del ciclo de vida y plantea la siguiente clasificación:

| TIPOS | | DESCRIPCIÓN |
|---|---|---|
| **De usuario** | **De capacidad** | Definen las funciones del software desde el punto de vista del usuario. |
| | **De restricción** | Establecen limitaciones a los requisitos de capacidad. |
| **De software** | **Funcionales** | Definen una función característica del futuro software. |
| | **De rendimiento** | Establecen límites al rendimiento y al volumen de información que el software debe tratar. |
| | **De interfaz** | Especifican los elementos hardware y/o software con los que el sistema se comunicará. |
| | **Operacionales** | Especifican cómo funcionará el software y cómo se comunicará con el usuario (diseño de la pantalla, estilo del lenguaje de comandos, etc.). |
| | **De recursos** | Especifican los límites de recursos físicos (memoria, espacio en disco, etc.). |

[16] European Space Agency

| | | |
|---|---|---|
| | **De verificación** | Definen características que facilitan la verificación de los requisitos. |
| | **De aceptación** | Requisitos para la fase de transferencia de software a las operaciones. |
| | **De documentación** | Definen el formato y estilo de la documentación que acompañará al software. |
| | **De seguridad** | Definen las características de control de acceso al software y otros aspectos relacionados con la seguridad del sistema y la información. |
| | **De portabilidad** | Especifican la facilidad del software para ser implantado en un entorno distinto. |
| | **De calidad** | Definen características que hacen al software apropiado para su propósito. |
| | **De fiabilidad** | Especifican el tiempo medio aceptable entre dos fallos producidos en el software. |
| | **De mantenimiento** | Definen la facilidad del software para ser modificado a la hora de corregir defectos, mejorar su ejecución, etc. |
| | **De protección** | Definen la seguridad de las personas frente a fallos del sistema. |

**Tabla 11. Clasificación de requisitos según estándares de la Agencia Espacial Europea.**

La caracterización de cada tipo de requisito se hace de manera muy similar a la que establece Métrica V3:

| ATRIBUTO | DESCRIPCIÓN |
|---|---|
| **Identificador** | Nombre del requisito, conciso y no ambiguo, que hace referencia a un único requisito del sistema. |
| **Necesidad** | Indica si se trata de un requisito esencial en el sistema, en cuyo caso no estarán sujetos a negociaciones. |
| **Prioridad** | Indica el orden temporal de realización del requisito. Puede tomar uno de siguientes valores: *alta, media, baja*. |
| **Estabilidad** | Se definen aquellos requisitos que deben ser estables a lo largo del ciclo de vida del proyecto y aquellos susceptibles de ser alterados debido al diseño arquitectónico o al diseño detallado y producción del código. |
| **Origen** | Hace referencia al documento externo, usuario o grupo de usuarios que establece un determinado requisito. |
| **Claridad** | Se incluyen especificaciones detalladas con objeto de evitar ambigüedades. |
| **Verificabilidad** | Incluye comprobaciones para evaluar que el requisito sea finalmente incorporado a la fase de diseño, así como implementado y probado. |

**Tabla 12. Definición de atributos de un requisito según estándares de la Agencia Espacial Europea.**

En nuestro caso, dividiremos la catalogación de requisitos en dos tipos, cada uno con sus diferentes subtipos:

- **Requisitos funcionales:**
  - **Requisitos de capacidad**

- **Requisitos no funcionales:**
  - **Requisitos de disponibilidad**
  - **Requisitos de interfaz**
  - **Requisitos de recursos**
  - **Requisitos de documentación**
  - **Requisitos de diseño**
  - **Requisitos de portabilidad**
  - **Requisitos de seguridad**
  - **Requisitos de implantación**

El formato finalmente escogido para la especificación de los requisitos de la plataforma de monitorización está descrito en la siguiente tabla:

| **IDENTIFICADOR:** | |
|---|---|
| **Nombre:** | |
| **Descripción:** | |
| **Necesidad:** | |
| **Prioridad:** | |
| **Estabilidad:** | |
| **Prerrequisito:** | |
| **Fuente:** | |

**Tabla 13. Formato para la especificación de requisitos.**

Donde:

- **Identificador**: identifica unívocamente a cada requisito. El formato escogido para representar cada requisito está descrito en la figura que sigue.


La cadena **"CCTV-MON"** se utiliza para indicar que todos los requisitos especificados en este documento hacen referencia al proyecto de monitorización de la red CCTV de la UC3M.

**Tipo:** incluirá tanto el grupo principal del requisito (funcional o no funcional) como el subtipo de requisito. Si es funcional, su valor será '*RF*' y '*RNF*' si es no funcional.

**Subtipo**: si se trata de un requisito funcional, su valor estará en la lista {CAP}, donde CAP = Capacidad. En caso de tratarse de un requisito no funcional, tomará uno de los valores de la lista {*IFC, REC, DOC, SEG, POR, MANT, DISP, IMP*}, donde IFC = De interfaz, REC = De recursos, DOC = De documentación, SEG = De seguridad, POR = De portabilidad, MANT = De mantenimiento, DISP = Disponibilidad y IMP = De implantación.

**Número:** número de tres cifras cuya secuencia comienza en "*001*".

- **Nombre**: nombre descriptivo que defina al requisito.
- **Descripción**: incluye una descripción textual clara y precisa del requisito.
- **Necesidad**: indica la necesidad del requisito en el sistema. Puede tomar uno de los valores de la lista {*Esencial, Deseable, Opcional*}.
- **Prioridad**: establece la prioridad del requisito dentro del catálogo de requisitos. Puede tomar un valor dentro de {*Alta, Media, Baja*}.
- **Estabilidad**: indica la probabilidad de cambio del requisito a lo largo del desarrollo del proyecto. Tomará un valor dentro de dentro de {*Alta, Media, Baja*}.
- **Prerrequisito**: es probable que el requisito necesite de otro requisito previo para poderse satisfacer. En ese caso se indicará en este campo de qué requisito se trata.
- **Fuente**: origen a partir del cual se extrae el requisito.

## a) Requisitos funcionales

| IDENTIFICADOR: CCTV-MON-RF-CAP-001 | |
|---|---|
| **Nombre:** | Equipos hardware a monitorizar |
| **Descripción:** | La herramienta de monitorización permitirá monitorizar cualquier equipo hardware que pueda identificarse con una dirección IP. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 14. Requisito CCTV-MON-RF-CAP-001**

| IDENTIFICADOR: CCTV-MON-RF-CAP-002 | |
|---|---|
| **Nombre:** | Elementos de monitorización |
| **Descripción:** | La herramienta de monitorización monitorizará elementos del *Hardware* y del *Software* (aplicaciones y servicios) de los equipos instalados en entorno en el que se implante la herramienta. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 15. Requisito CCTV-MON-RF-CAP-002**

| IDENTIFICADOR: CCTV-MON-RF-CAP-003 | |
|---|---|
| **Nombre:** | Inserción de equipos a monitorizar |
| **Descripción:** | La herramienta de monitorización permitirá la inserción manual de equipos a monitorizar. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 16. Requisito CCTV-MON-RF-CAP-003**

| IDENTIFICADOR: CCTV-MON-RF-CAP-004 | |
|---|---|
| **Nombre:** | Reglas de descubrimiento |
| **Descripción:** | La inserción de equipos a monitorizar se podrá realizar de forma automática a través de reglas de descubrimiento definidas sobre una red específica. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Baja |
| **Estabilidad:** | Media. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 17. Requisito CCTV-MON-RF-CAP-004**

| IDENTIFICADOR: CCTV-MON-RF-CAP-005 | |
|---|---|
| **Nombre:** | Actualización de equipos monitorizados |
| **Descripción:** | La herramienta de monitorización permitirá actualizar los datos de los equipos *Hardware* monitorizados así como borrar estos últimos. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RF-CAP-003, CCTV-MON-RF-CAP-004 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 18. Requisito CCTV-MON-RF-CAP-005**

| IDENTIFICADOR: CCTV-MON-RF-CAP-006 | |
|---|---|
| **Nombre:** | Inserción de elementos de monitorización |
| **Descripción:** | La herramienta de monitorización posibilitará la inserción manual de elementos *Hardware* y *Software* de los equipos que estén siendo monitorizados. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RF-CAP-003, CCTV-MON-RF-CAP-004 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 19. Requisito CCTV-MON-RF-CAP-006**

| IDENTIFICADOR: CCTV-MON-RF-CAP-007 | |
|---|---|
| **Nombre:** | Actualización de elementos de monitorización |
| **Descripción:** | La herramienta de monitorización ofrecerá la funcionalidad de modificación de los elementos de monitorización ya existentes así como el borrado de los mismos. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RF-CAP-006 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 20. Requisito CCTV-MON-RF-CAP-007**

| IDENTIFICADOR: CCTV-MON-RF-CAP-008 | |
|---|---|
| **Nombre:** | Creación de alertas |
| **Descripción:** | La herramienta de monitorización brindará la posibilidad de crear alertas a través de las cuales se notificará el estado del sistema. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 21. Requisito CCTV-MON-RF-CAP-008**

| IDENTIFICADOR: CCTV-MON-RF-CAP-009 | |
|---|---|
| **Nombre:** | Niveles de las alertas |
| **Descripción:** | Las alertas creadas tendrán diferentes niveles en función de la criticidad del evento que notifican. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RF-CAP-008 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 22. Requisito CCTV-MON-RF-CAP-009**

| IDENTIFICADOR: CCTV-MON-RF-CAP-010 | |
|---|---|
| **Nombre:** | Notificación de las alertas |
| **Descripción:** | El estado y nivel de la alerta así como el evento asociado se podrán notificar a través del envío de correo electrónico, SMS o mensajería instantánea. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RF-CAP-008 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 23. Requisito CCTV-MON-RF-CAP-010**

| IDENTIFICADOR: CCTV-MON-RF-CAP-011 | |
|---|---|
| **Nombre:** | Gráficos de monitorización |
| **Descripción:** | La herramienta de monitorización permitirá obtener y definir gráficos en los que se refleje el estado a lo largo del tiempo de los elementos monitorizados. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 24. Requisito CCTV-MON-RF-CAP-011**

| IDENTIFICADOR: CCTV-MON-RF-CAP-012 | |
|---|---|
| **Nombre:** | Creación de mapas y pantallas |
| **Descripción:** | En la herramienta se podrán configurar mapas y pantallas en los que mostrar la infraestructura del sistema o un resumen de su estado con los datos más importantes de la monitorización. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Baja |
| **Estabilidad:** | Alta. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 25. Requisito CCTV-MON-RF-CAP-012**

| IDENTIFICADOR: CCTV-MON-RF-CAP-013 | |
|---|---|
| **Nombre:** | Comandos remotos |
| **Descripción:** | La herramienta permitirá configurar comandos remotos que se ejecutarán sobre un determinado equipo cuando se produzca una alerta de un determinado nivel. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Baja |
| **Estabilidad:** | Media. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 26. Requisito CCTV-MON-RF-CAP-013**

| IDENTIFICADOR: CCTV-MON-RF-CAP-014 | |
|---|---|
| **Nombre:** | Almacenamiento de los datos de monitorización |
| **Descripción:** | Los datos recogidos en la monitorización se almacenarán en una base de datos con objeto de poder conservar un histórico que sirva de base para futuros estudios y comparativas. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 27. Requisito CCTV-MON-RF-CAP-014**

| IDENTIFICADOR: CCTV-MON-RF-CAP-015 | |
|---|---|
| **Nombre:** | Creación de informes |
| **Descripción:** | Los datos de monitorización podrán presentarse en informes de distintos tipos y configurables según varios criterios. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Media |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 28. Requisito CCTV-MON-RF-CAP-015**

| IDENTIFICADOR: CCTV-MON-RF-CAP-016 | |
|---|---|
| **Nombre:** | Creación de scripts |
| **Descripción:** | Se podrán crear en la herramienta scripts a través de los cuales ejecutar una serie de comandos sobre los equipos monitorizados. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Media |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 29. Requisito CCTV-MON-RF-CAP-016**

| IDENTIFICADOR: CCTV-MON-RF-CAP-017 | |
|---|---|
| **Nombre:** | Agentes en los equipos monitorizados |
| **Descripción:** | La herramienta de monitorización se comunicará con los equipos monitorizados a través de un agente software instalado en ellos. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 30. Requisito CCTV-MON-RF-CAP-017**

| IDENTIFICADOR: CCTV-MON-RF-CAP-018 | |
|---|---|
| **Nombre:** | Soporte SNMP |
| **Descripción:** | La herramienta de monitorización contará con soporte para el protocolo SNMP. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 31. Requisito CCTV-MON-RF-CAP-018**

## b) Requisitos no funcionales

### 1. Requisitos de disponibilidad

| IDENTIFICADOR: CCTV-MON-RNF-DISP-001 | |
|---|---|
| **Nombre:** | Disponibilidad de la herramienta de monitorización |
| **Descripción:** | La herramienta de monitorización deberá estar disponible las 24 horas del día y durante 7 días a la semana. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 32. Requisito CCTV-MON-RNF-DISP-001**

### 2. Requisitos de interfaz

| IDENTIFICADOR: CCTV-MON-RNF-IFC-001 | |
|---|---|
| **Nombre:** | Comunicación del usuario con la herramienta de monitorización |
| **Descripción:** | El usuario podrá acceder a la herramienta de monitorización e interactuar con ésta a través de un cliente Web que utilizará el protocolo HTTP. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 33. Requisito CCTV-MON-RNF-IFC-001**

| IDENTIFICADOR: CCTV-MON-RNF-IFC-002 | |
|---|---|
| **Nombre:** | Persistencia de los datos de monitorización |
| **Descripción:** | Los datos obtenidos en el proceso de monitorización se mantendrán en una base de datos. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 34. Requisito CCTV-MON-RNF-IFC-002**

## 3. Requisitos de recursos

| IDENTIFICADOR: CCTV-MON-RNF-REC-001 | |
|---|---|
| **Nombre:** | Entorno *hardware* de la herramienta de monitorización |
| **Descripción:** | "*Zabbix-cctv*", la máquina donde se instala la herramienta de monitorización, es un servidor HP Proliant DL360 G4 con un procesador Intel Xeon a Ghz, 2 discos Ultra SCSI de 300 Gb de capacidad y 2 Gb de memoria RAM. Dispone de 2 interfaces de red. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 35. Requisito CCTV-MON-RNF-REC-001**

## 4. Requisitos de documentación

| IDENTIFICADOR: CCTV-MON-RNF-DOC-001 | |
|---|---|
| **Nombre:** | Documentación de la plataforma de monitorización |
| **Descripción:** | Al concluir las fases de análisis, diseño y despliegue de la plataforma de monitorización se elaborará una documentación en la que se presenten las actividades llevadas a cabo. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 36. Requisito CCTV-MON-RNF-DOC-001**

| IDENTIFICADOR: CCTV-MON-RNF-DOC-002 | |
|---|---|
| **Nombre:** | Manual de operación de la herramienta |
| **Descripción:** | Una vez desplegada e implantada la plataforma de monitorización, se elaborará un manual en el que se describa la manera de operar con la herramienta Zabbix. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Baja |
| **Estabilidad:** | Media. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 37. Requisito CCTV-MON-RNF-DOC-002**

# 5. Requisitos de diseño

| IDENTIFICADOR: CCTV-MON-RNF-DIS-001 | |
|---|---|
| **Nombre:** | Sistema operativo de la plataforma de monitorización |
| **Descripción:** | El sistema operativo del equipo en que se instalará la herramienta de monitorización será GNU/Linux. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 38. Requisito CCTV-MON-RNF-DIS-001**

| IDENTIFICADOR: CCTV-MON-RNF-DIS-002 | |
|---|---|
| **Nombre:** | Contenedor Web de la plataforma de monitorización |
| **Descripción:** | Se utilizará Apache 2.2 como contenedor Web. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 39. Requisito CCTV-MON-RNF-DIS-002**

| IDENTIFICADOR: CCTV-MON-RNF-DIS-003 | |
|---|---|
| **Nombre:** | Servidor de bases de datos de la plataforma de monitorización |
| **Descripción:** | El gestor de bases de datos utilizado será MySQL con versión 5.1. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 40. Requisito CCTV-MON-RNF-DIS-003**

| IDENTIFICADOR: CCTV-MON-RNF-DIS-004 | |
|---|---|
| **Nombre:** | Paquetes SNMP |
| **Descripción:** | Para el soporte al protocolo SNMP se utilizarán los paquetes oficiales disponibles. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 41. Requisito CCTV-MON-RNF-DIS-004**

## 6. Requisitos de portabilidad

| IDENTIFICADOR: CCTV-MON-RNF-POR-001 | |
|---|---|
| **Nombre:** | Portabilidad de la plataforma |
| **Descripción:** | La plataforma de monitorización será portable a cualquier equipo del sistema de videovigilancia que cuente con un sistema operativo GNU/Linux. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 42. Requisito CCTV-MON-RNF-POR-001**

## 7. Requisitos de seguridad

| IDENTIFICADOR: CCTV-MON-RNF-SEG-001 | |
|---|---|
| **Nombre:** | Acceso mediante autenticación |
| **Descripción:** | El acceso a la interfaz web de la herramienta de monitorización se realizará a través de un usuario y contraseña. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 43. Requisito CCTV-MON-RNF-SEG-001**

| IDENTIFICADOR: CCTV-MON-RNF-SEG-002 | |
|---|---|
| **Nombre:** | Creación de usuarios |
| **Descripción:** | La herramienta de monitorización permitirá la creación de usuarios de la aplicación. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 44. Requisito CCTV-MON-RNF-SEG-002**

| IDENTIFICADOR: CCTV-MON-RNF-SEG-003 | |
|---|---|
| **Nombre:** | Creación de grupos de usuarios |
| **Descripción:** | La herramienta de monitorización ofrecerá la opción de crear grupos en los que incluir a los usuarios creados individualmente. |
| **Necesidad:** | Deseable |
| **Prioridad:** | Media |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | CCTV-MON-RNF-SEG-002 |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 45. Requisito CCTV-MON-RNF-SEG-003**

| IDENTIFICADOR: CCTV-MON-RNF-SEG-004 | |
|---|---|
| **Nombre:** | Privilegios de usuarios |
| **Descripción:** | En la herramienta de monitorización se podrán asignar privilegios funcionales a nivel de usuario o de grupos de usuario. |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 46. Requisito CCTV-MON-RNF-SEG-004**

# 8. Requisitos de implantación

| IDENTIFICADOR: CCTV-MON-RNF-IMP-001 | |
|---|---|
| **Nombre:** | Costes de implantación |
| **Descripción:** | El coste económico en términos de software que supondrá la implantación de la plataforma será nulo, por lo que la herramienta de monitorización deberá ser gratuita (*Open Source*). |
| **Necesidad:** | Esencial |
| **Prioridad:** | Alta |
| **Estabilidad:** | Alta. Durante toda la vida del sistema. |
| **Prerrequisito:** | Ninguno |
| **Fuente:** | Equipo de seguridad informática de la UC3M |

**Tabla 47. Requisito CCT-MON-RNF-IMP-001**

# 5

# Diseño de la plataforma de monitorización

# 5. DISEÑO DE LA PLATAFORMA DE MONITORIZACIÓN

En este apartado se explican las decisiones de diseño sobre las que se basará la plataforma de monitorización para satisfacer los requisitos definidos en la fase de análisis.

## 5.1. Elección de la herramienta de monitorización

Tal como vimos en el apartado 3 (**Estado de la cuestión**), existen diversas herramientas de monitorización *Open Source* destinadas a la monitorización de infraestructuras, todas ellas fácilmente ajustables al sistema de videovigilancia en el que nos centramos en este proyecto.

Cuando inicialmente se tomó la decisión de crear la plataforma de monitorización, fueron varias las herramientas software que se examinaron hasta encontrar un candidato final.

Se consideraron inicialmente 3 soluciones: Zabbix, Nagios y Pandora. Ya hemos tratado sobre éstas en secciones previas (Estado de la cuestión) por lo que aquí nos limitaremos a señalar los motivos por los que finalmente fue Zabbix la elección adoptada.

En esta sección mostraremos una comparativa entre esas 3 herramientas, tras la cual se seleccionará una de ellas exponiendo los motivos que nos llevan a tomar esa decisión. Para llevar a cabo la comparativa se creó una instalación piloto de esas 3 herramientas, y con cada una de esas instalaciones se ejecutarán distintos casos de prueba o de uso con los que mostrar los aspectos destacables y mejorables de cada una de las soluciones software estudiadas.

### 5.1.1. Equivalencias conceptuales entre Zabbix, Nagios y Pandora FMS

A modo de introducción de esta comparativa, mostraremos cómo implementa cada solución software los conceptos teóricos básicos que son comunes a todos ellas.

El primer punto que comparten es el concepto de *host* o equipo a monitorizar y Zabbix y Nagios lo denominan tal cual, a diferencia de Pandora FMS, que los identifica como *agent*.

Obviamente en las tres herramientas evaluadas se monitorizan una serie de parámetros de cada host. Para Zabbix, esos parámetros reciben el nombre de *ítems*; en el caso de Nagios, el equivalente de un *ítem* se denomina *service*, y, por último, Pandora FMS lo llama *módulo o module*.

El siguiente aspecto es el cómo se obtienen los valores correspondientes a cada uno de esos parámetros de monitorización. El modo de recopilación de valores en Zabbix se basa en una clave del ítem que identifica el parámetro en cuestión; en Nagios, para cada *service*, se ejecuta un comando basado en un script que recibe como argumento una variable referida al parámetro de monitorización (p.ej. CPULOAD, UPTIME); en Pandora FMS existen varias técnicas para la obtención de los valores de monitorización, y la que se ha probado en esta comparativa se basa en el protocolo *Tentacle* **[30]**, que lee los datos de un fichero xml creado por el agente de Pandora.

Con los datos de monitorización ya recopilados, cada herramienta tiene su particular mecanismo de notificar posibles alertas (y umbrales de alerta) sobre esos valores. En Zabbix existen *triggers* o iniciadores que se activan según el parámetro de monitorización tome un cierto valor o sobrepase un determinado umbral. Nagios lo implementa de forma similar, a diferencia de que no existen disparadores como tal, sino que en la ejecución del comando del service recibe dos argumentos; uno para el nivel de alerta *Warning* (-w) y otro para el nivel de alerta *Critical* (-c). Para más información sobre las posibles formas de configurar los umbrales de alerta de Nagios, ver referencia **[31]**. Por último, Pandora FMS asocia a cada parámetro monitorizado niveles de alerta cuyos rangos o umbrales son configurables por el usuario (por defecto, los estados posibles son *Correcto, Advertencia/aviso, Crítico, Desconocido, No inicializado*).

Tanto Zabbix como Nagios y Pandora FMS incorporan mecanismos con los que reaccionar frente a ciertos niveles de alerta registrados. En el caso de Zabbix podemos asociar *acciones* que se ejecutarán como resultado de la activación de un disparador o *trigger*. Estas acciones pueden ser de 2 tipos (envío de mensaje o ejecución de comando remoto). Nagios dispone de notificaciones a las que se le asocia la ejecución de un comando o script y, finalmente, Pandora FMS puede ejecutar un comando que se encargue de notificar la alerta a través de varios canales, como por ejemplo la escritura en un fichero de log, el envío de un correo electrónico o de un mensaje SMS.

En la siguiente tabla mostramos un resumen de los conceptos equivalentes que implementan Zabbix, Nagios y Pandora FMS:

| Concepto | ZABBIX | Nagios | Pandora FMS |
|---|---|---|---|
| Parámetro de monitorización | ITEM | SERVICE | MODULE |
| Modo de ejecución para obtención de parámetros de monitorización | CLAVE EN EL ITEM | COMANDO EN EL SERVICE | FICHERO XML |
| Equipo a monitorizar | HOST | HOST | AGENT |
| Indicadores de alerta | TRIGGER | CONDICIONES EN LA EJECUCIÓN DEL COMANDO DEL SERVICE | NIVELES DE ALERTA ASOCIADOS AL PARÁMETRO MONITORIZADO Y CONFIGURABLES POR EL USUARIO |
| Acciones ejecutadas ante una alerta | ACTION | NOTIFICATION | COMMAND |
| Mecanismos de notificación de alertas | ENVÍO DE MENSAJES, COMANDOS REMOTOS | SCRIPTS PERSONALIZABLES | ENVÍO DE MENSAJES (e-mail, SMS), LOGS, SONIDOS |

**Tabla 48. Resumen de equivalencias entre Zabbix, Nagios y Pandora FMS**

## 5.2. Ejemplos de casos de uso

En este punto mostraremos una serie de casos de uso con los que apreciar las diferencias en cuanto a configuración de cada una de las 3 herramientas de monitorización estudiadas.

### 5.2.1. Crear y configurar un host

❶ ZABBIX

La creación de un host en Zabbix puede realizarse desde el propio frontend. Para configurarlo basta con introducir el nombre del host, su dirección IP y, opcionalmente, un grupo dentro del que incluirlo y una plantilla con ítems que contengan parámetros de monitorización. También es posible configurar reglas de auto-descubrimiento para que sea el propio Zabbix quien introduzca automáticamente los equipos.

**Ventajas:** la introducción de datos a través del frontend web simplifica la operación.

**Inconvenientes:** el nombre que asignemos al host debe coincidir con el nombre que hayamos escrito en el fichero de configuración del agente instalado en dicho equipo.

❷ Nagios

En este caso la creación de un host debe hacerse editando los ficheros de configuración y siguiendo una determinada sintaxis:

```
define host{
        use             windows-server  ; Plantilla de la que se heredan parámetros
        host_name       name   ; Nombre del host
        alias           alias    ; Nombre descriptivo del host
        address         192.168.14.61 ; Dirección IP del host

    }
```

**Ventajas:** permite la "herencia" de parámetros entre host mediante la definición de plantillas. Si definimos un grupo o host con un nombre "A", los host que en los que declaremos posteriormente la opción "use A" heredarán automáticamente las propiedades del host "A". Esto evita, en caso de tener un grupo de host homogéneos, tener que reescribir la misma información varias veces.

**Inconvenientes:** exige editar los ficheros de configuración manualmente (operación más tediosa que a través de un frontend web) y, para ello, se debe conocer previamente la ubicación de los mismos. Además, esta técnica es susceptible a errores de sintaxis por parte del usuario. No existen reglas de auto-descubrimiento con las que ahorrar trabajo al usuario final. Cada vez que se crea un nuevo *host*, se debe reiniciar el servidor Nagios para que éste recoja los cambios.

❸ Pandora FMS

Al igual que en Zabbix, la creación de equipos puede realizarse a través del frontend o por medio de reglas de descubrimiento configurables también desde el frontend web.

**Ventajas:** la introducción de datos a través del frontend web simplifica la operación. Se puede definir el sistema operativo del host y, en el caso de las reglas de auto-descubrimiento, Pandora FMS puede detectar el tipo de dispositivo.

**Inconvenientes:** a la hora de crear el host, ciertos parámetros (como el modo de definición de módulos) no quedan claros al usuario.

### 5.2.2. Crear un parámetro de monitorización

❶ ZABBIX

Para crear un parámetro de monitorización o *item* en Zabbix lo podemos hacer a nivel de host o a nivel de plantilla. En ambos casos lo haremos desde el mismo menú del frontend web (*Configuration -> Hosts -> Opción Items en el menú desplegable -> Create Item*), con la diferencia de que, cuando lo hagamos a nivel de plantilla, debemos especificar el nombre de la plantilla en el campo "*Host*" del formulario que se nos presentará.

**Ventajas:** la introducción de datos a través del frontend web simplifica la operación. La posibilidad de definir plantillas de *ítems* es útil cuando se tienen grupos de host homogéneos. Por defecto, Zabbix incluye un gran número de valores para el parámetro *key* (clave del ítem) y permite que el usuario defina además, sus propios scripts (*user parameters* **[32]**).

**Inconvenientes:** el host o plantilla al que asignemos el *ítem* tiene que estar necesariamente ya introducido en la herramienta antes de crear el *item*.

❷ Nagios

Al igual que en la creación de un host, la definición de un *service* o parámetro de monitorización en Nagios se debe hacer editando el correspondiente fichero de configuración y siguiendo una cierta sintaxis:

```
define service{
        use                     generic-serv ; Plantilla de la que se heredan parámetros
        host_name               name ; Nombre del servicio
        service_description     Description ; Nombre descriptive del servicio
        check_command           check_nt!UPTIME!-w 600: ; Comando a ejecutar

    }
```

**Ventajas:** permite la "herencia" de parámetros entre *services* mediante la definición de plantillas. Si definimos un *service* con un nombre "A", los demás *services* en los que declaremos posteriormente la opción "use A" heredarán automáticamente las propiedades del service "A". Esto evita, en caso de tener un grupo de *services*

homogéneos, tener que reescribir la misma información varias veces. El comando a ejecutar puede ser cualquier script definido por el usuario.

**Inconvenientes:** exige editar los ficheros de configuración manualmente (operación más tediosa que a través de un frontend web) y, para ello, se debe conocer previamente la ubicación de los mismos. Además, esta técnica es susceptible a errores de sintaxis por parte del usuario. El conjunto de variables que pueden recibir como argumento los plugins que incluye Nagios por defecto (check_nt, check_snmp…) es un tanto restringido, a diferencia de Zabbix. Cada vez que se crea un nuevo *service*, se debe reiniciar el servidor de Nagios para que éste recoja los cambios.

❸ Pandora FMS

Para asignar a un host parámetros de monitorización o *modules* en Pandora FMS podemos editar el fichero de configuración de cada agente software o bien utilizar el menú del frontend web que proporciona Pandora (*Administration -> Manage modules*). En este último caso no podremos crear nuevos parámetros sino utilizar los ya existentes.

**Ventajas:** la introducción de datos a través del frontend web simplifica la operación. Al igual que en Zabbix también es posible la definición de plantillas que incluyan diversos parámetros de monitorización.

**Inconvenientes:** si deseamos introducir nuevos *modules* a través del frontend web deberemos adquirir la versión *Enterprise* de Pandora FMS (no gratuita).

### 5.2.3. Crear una alerta

❶ ZABBIX

En Zabbix las alertas se crean en base a los disparadores o *triggers*, los cuales evalúan el valor de un cierto parámetro de monitorización y se activan en caso de que dicho valor esté dentro de o supere unos límites definidos por el usuario. Como en el resto de casos de uso, crearemos el *trigger* desde el menú correspondiente del frontend web (*Configuration -> Hosts -> Opción Triggers en el menú desplegable -> Create Trigger*). El disparador puede ser asignado a un host o bien a una plantilla, tal como se hace a la hora de crear un *ítem*.

**Ventajas:** la introducción de datos a través del frontend web simplifica la operación. La posibilidad de incluir los *triggers* en plantillas es útil cuando se tienen grupos de host homogéneos. No es necesario conocer la sintaxis de creación de disparadores pues el propio Zabbix construye el disparador con las opciones que se le indican. Es

posible definir dependencias entre disparadores para que así el sistema no registre alertas innecesarias.

**Inconvenientes:** se puede personalizar el nivel de "severidad" del *trigger* con 5 posibles niveles (*Average, High, Information, Warning, Disaster*), lo que añade complejidad innecesaria.

❷ Nagios

En Nagios no es necesario configurar disparadores ya que en el propio *service* se pueden definir criterios de alerta:

```
define service{
        use                     generic-serv ; Plantilla de la que se heredan parámetros
        host_name               name ; Nombre del servicio
        service_description     Description ; Nombre descriptive del servicio
        check_command           check_nt!UPTIME!-w 600: ; Comando a ejecutar

    }
```

En este ejemplo de *service* podemos ver que se ha especificado en el comando la opción "-w 600:", que indica que saltará una alerta de WARNING cuando el valor de la variable UPTIME sea menor a 600. También es posible definir el nivel de alerta CRITICAL con la opción "-c" seguida de un valor. Para saber cómo configurar los umbrales de alerta de Nagios, ver la referencia [31].

**Ventajas:** la definición de alertas es tan simple como añadir dos argumentos al comando que se ejecuta en cada *service*.

**Inconvenientes:** a pesar de que son varias las opciones para modificar los umbrales de alerta, tan sólo hay dos niveles (WARNING y CRITICAL).

❸ Pandora FMS

En Pandora FMS las alertas están compuestas por dos conceptos relacionados entre sí. Por un lado existen **comandos** predefinidos (el usuario puede crear más comandos personalizados) y, por otro, las **acciones**. De manera similar a Zabbix, las acciones enlazan los comandos con parámetros de monitorización específicos.

**Ventajas:** el listado de comandos que Pandora FMS puede ejecutar es flexible, permitiendo añadir más comandos a los que se incluyen por defecto.

**Inconvenientes:** cada acción se puede vincular solamente a un único comando.

### 5.2.4. Crear acciones asociadas a una alerta

❶ ZABBIX

Una vez se activa un disparador, se pueden crear acciones asociadas con las que se pueden notificar el estado de la alerta generada con el disparador. Desde el menú *Configuration -> Actions* se crean las acciones con unas condiciones (disparador, valor devuelto por el disparador, host en el que se ha generado la alerta, etc.) y unas operaciones a ejecutar, como por ejemplo el envío de un correo electrónico cuando se genere el evento y cuando éste haya quedado solucionado.

**Ventajas:** la creación de acciones se realiza a través del frontend web de manera sencilla. Las condiciones de cada acción son flexibles y se pueden relacionar a nivel lógico entre ellas (AND/OR). Existe un amplio catálogo de macros (ver ANEXO III) con el que escribir información muy útil en el cuerpo de los mensajes de correo electrónico que se envíen como notificación. Se pueden programar

**Inconvenientes:** la acción creada no funcionará si el usuario al que se envían los mensajes no tiene permisos de al menos lectura sobre el host en el que se ha generado el evento.

❷ Nagios

Las acciones se programan editando el fichero de configuración de los comandos que ejecuta el servidor. La sintaxis de un ejemplo de notificación es la siguiente:

```
define command{

        command_name    notify-host-by-email
        command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification
                        Type: $NOTIFICATIONTYPE$\nHost:
                        $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
                        $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time:
                        $LONGDATETIME$\n" | /usr/bin/mail -s "**
                        $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is
                        $HOSTSTATE$ **" $CONTACTEMAIL$

    }
```

Se crea un comando con un nombre y un script a ejecutar. Dicho script acepta, al igual que las acciones de Zabbix, la inclusión de macros con las que representar información del host y tipo de alerta generada.

**Ventajas:** el comando a ejecutar para cada acción se puede parametrizar con cualquier script.

**Inconvenientes:** nuevamente se hace necesario editar un fichero de configuración y pueden producirse errores de sintaxis a la hora de definir los atributos de la notificación.

❸ Pandora FMS

Tal como hemos indicado en el punto anterior, las acciones son uno de los componentes que conforman las alertas en Pandora y para su creación se sigue un procedimiento similar al seguido en Zabbix.

### 5.2.5. Crear un parámetro de monitorización SNMP

❶ ZABBIX

La creación de parámetros de monitorización obtenidos a través de SNMP es prácticamente igual a la creación de cualquier otro parámetro de monitorización o *item*. La diferencia radica en que, en el campo *type* del item se debe indicar que se trata de un agente SNMP, ya sea en cualquiera de las 3 versiones del protocolo. El atributo más importante para crear un *item SNMP* es el OID, el cual podemos obtener ejecutando en el propio servidor de Zabbix la utilidad *snmpwalk* **[33]**.

**Ventajas:** es posible elegir entre las 3 versiones del procotolo SNMP.

**Inconvenientes:** se necesita una utilidad externa para obtener los identificadores OID necesarios para crear los *items SNMP*.

❷ Nagios

Como el resto de parámetros o *services*, los valores obtenidos a través de SNMP se declaran en los ficheros de configuración. La sintaxis seguida es la misma que para cualquier otro *service* con la salvedad de que ahora el comando a ejecutar es el plugin *check_snmp* **[34]**, el cual recibe como argumento el OID del objeto SNMP a monitorizar.

```
define service{
        use                     generic-serv ; Plantilla de la que se heredan parámetros
        host_name               name ; Nombre del servicio
        service_description     Description ; Nombre descriptive del servicio
        check_command           check_snmp!(OID del objeto SNMP); Comando a ejecutar

    }
```

**Ventajas:** la creación del parámetro es muy simple. Tan sólo es necesario conocer el OID para obtener el valor correctamente. El resto de ventajas son exactamente las mismas que para el resto de parámetros de monitorización.

**Inconvenientes:** al igual que en el resto de parámetros, se debe editar el fichero de configuración sin otra alternativa posible.

❸ Pandora FMS

La monitorización de parámetros SNMP en Pandora FMS se realiza a través de *traps SNMP* (un trap es un mensaje enviado por un equipo). Para gestionar esos traps, Pandora FMS dispone, en su frontend web, de una consola desde la que gestionar esos mensajes. También es posible definir módulos SNMP editando los correspondientes ficheros de configuración.

**Ventajas:** a la hora de definir el trap SNMP es posible asignarle una acción que se ejecutará una vez el servidor SNMP de Pandora FMS analice el valor recibido.

**Inconvenientes:** se trata de una operación un tanto complicada, pues exige previamente crear un "agente SNMP" en Pandora.

## 5.3. Conclusiones finales

En primer lugar, incluimos un listado con las valoraciones obtenidas tras la utilización de cada una de las 3 herramientas software:

### ❶ Nagios.

- **A destacar:**
  - *Open Source*.
  - Existencia de una gran comunidad de usuarios.
  - Funcionalidad *soft states/hard states* [35] para evitar falsas alarmas.
  - Simplicidad en la estructura de funcionamiento (*host + service + command*)
  - Las alertas están incluidas dentro del comando que ejecuta el *service* (opciones –w y –c)
  - Gran cantidad de potentes plugins con facilidad para crear plugins por parte del usuario.
  - Representación simplificada del estado de un parámetro de monitorización (*estado + resumen + datos*)
  - Frontend web fácil de utilizar.
  - Personalización del intervalo de ejecución de checks a través del frontend web.
  - Los plugins de depuración son relativamente simples.
  - Disponibilidad de aplicaciones para migrar Nagios a otras soluciones de monitorización.
  - Funcionalidades ingeniosas como los grupos de host u opciones de notificaciones.
  - Dependencias entre alertas para evitar el envío masivo de notificaciones innecesarias.
  - Monitoriza los principales protocolos (HTTP, FTP, SSH, SMTP, POP3, SNMP, etc.)
  - Plugin *nagvis* [11] para la creación de gráficos con los que visualizar los datos de monitorización.
  - Es una herramienta de monitorización escalable.
  - No necesita de ninguna otra aplicación para funcionar.

- **A mejorar:**
  - Su funcionamiento se centra en comprobaciones de disponibilidad. No se pueden obtener gráficos con los valores monitorizados, por lo que se precisa de una herramienta auxiliar con las que crearlos.

- La mayoría de parámetros de monitorización se configuran editando ficheros de configuración y no a través de la interfaz web.
- Se debe reiniciar el servidor Nagios cada vez que se inserta o modifica un nuevo parámetro de monitorización.
- Difícil en términos de aprendizaje de uso.
- Exige mayor trabajo por parte del usuario para su configuración y mantenimiento.
- No ofrece grandes prestaciones para la creación de gráficos.
- Para disponer de datos estadísticos (*trends*) es necesario instalar plugins adicionales.
- Necesita tener acceso vía SSH si se desea monitorizar parámetros internos del equipo.
- La configuración a nivel textual (ficheros de configuración) tiene demasiados parámetros, por lo que será habitual tener que revisarlos. Una configuración basada en una interfaz web resolvería estos problemas.
- Los plugins de terceros suelen estar pobremente programados y apenas disponen de documentación.
- Algunas vistas de la interfaz web no son intuitivas.
- Muchos plugins no tienen su correspondiente entrada en la configuración, así que es necesario averiguar cómo funcionan y escribir las configuraciones sobre sus propios ficheros de configuración, lo cual complica su uso para usuarios noveles en el manejo de la herramienta.
- Cada uno de los conjuntos de parámetros de un plugin necesita una entrada de configuración distinta.
- La mayor parte de las comprobaciones se ejecutan del lado del servidor de Nagios, lo cual no es precisamente óptimo y supone una elevada carga para dicho servidor.
- Por defecto, cada alerta supone una notificación. En ese caso, si no es posible definir dependencias de forma adecuada, se enviarán notificaciones en exceso.

### ❷ Pandora FMS.

- **A destacar:**
  - *Open Source*.
  - Fácil instalación y fácil configuración.
  - Costes bajos y afrontables. No requiere el pago de licencias adicionales (a excepción del módulo *Enterprise*).
  - Interfaz web a través de la cual se puede obtener de un vistazo el estado de la monitorización.

- Pre-configuración de distintos umbrales para los valores de monitorización, comparados continuamente con los valores actuales para poder responder proactivamente a los problemas notificados por las alertas.
- Servicio de informes basado en web.
- Integración de un sistema de gestión de incidencias (tickets).
- Reglas de descubrimiento que detectan distintos parámetros del equipo monitorizado, como el tipo de dispositivo de que se trata o su sistema operativo (si lo tuviera).
- Creación automática de gráficos indicando la evolución de los datos de la monitorización.
- Disponibilidad de distintos protocolos (SSH, FTP, Tentacle) para la transmisión de datos de monitorización desde los agentes al servidor.
- Facilita la monitorización de sistemas Windows a través de WMI **[38]**.
- Gran escalabilidad.
- Proporciona análisis de datos históricos y estadísticas (*trends*).
- Soporta un amplio catálogo de protocolos, entre los cuales se encuentran HTTP, FTP, SSH, SMTP, POP3 y SNMP.

- **A mejorar:**
  - Complejidad añadida sobre otras herramientas como Nagios.
  - Gran cantidad de los elementos a monitorizar se establecen a través de módulos que se definen de forma textual en los ficheros de configuración de los agentes.
  - No existe una amplia comunidad de usuarios de la herramienta.
  - Al ser una herramienta de reciente creación, se dispone de reducidas fuentes de documentación.
  - Las funcionalidades adicionales de la versión *Enterprise* no son gratuitas.
  - Los datos recopilados no se almacenan en una base de datos.

### ❸ Zabbix. ZABBIX

- **A destacar:**
  - *Open Source*.
  - Intuitivo y sencillo de aprender.
  - Monitorización a través de disponibilidad (checks) como en Nagios y a través de gráficos como en Pandora, todo ello en una única herramienta.
  - Altamente configurable.
  - Las alertas son concisas, con lo que realmente ayudan a quien recibe el mensaje de notificación.

- Altas prestaciones. Los agentes Zabbix pueden ser instalados en los sistemas y recopilar datos de forma eficiente en cada equipo integrado en dicho sistema. Incluso pueden ejecutar scripts para obtener información.
- Configuración de los intervalos de recopilación de datos. No es necesario esperar minutos hasta ver resultados. Cada elemento monitorizado tiene su propio intervalo de actualización.
- Interfaz web rápida.
- La configuración de los parámetros de monitorización se realiza enteramente a través de la interfaz o *frontend* web.
- Sofisticada monitorización de sitios web. Zabbix puede seguir una secuencia de clicks de ratón simulados en un sitio web y comprobar la funcionalidad y el tiempo de respuesta.
- Gráficos en tiempo real.
- Políticas de permisos a nivel de usuario de la herramienta.
- Los datos de monitorización se almacenan en una base de datos (MySQL, PostgreSQL, SQLite). El tiempo que los datos permanecen en la base de datos (historial) es igualmente configurable.
- Las plantillas (grupos de elementos de monitorización) ahorran tiempo a la hora de configurar múltiples comprobaciones.
- Creación de gráficos de forma automática para los elementos monitorizados en la última hora, la última semana y el último mes.
- Es posible crear gráficos para cualquier tipo de elemento de monitorización cuyo tipo de valor sea numérico.
- Los gráficos pueden personalizarse según el/los elemento/s de monitorización que presenten.
- Se pueden crear pantallas y diapositivas para crear vistas a mayor nivel, combinando elementos textuales y gráficos.
- Posibilidad de definir scripts para las alertas y notificaciones.
- Monitorización remota a través de los proxy de Zabbix.
- Disponibilidad de soporte y programación personalizados, ambos de pago.
- Documentación extensa en forma de manuales y referencias.
- Configuración de dependencias entre alertas para evitar el envío innecesario de notificaciones.
- Soporte de un amplio listado de protocolos (HTTP, FTP, SSH, SMTP, POP3, SNMP).
- El soporte de SNMP facilita, entre otras cosas, la monitorización de equipos en los que no se puede instalar el agente.
- No es necesario reiniciar el servidor cada vez que se modifica un parámetro de monitorización.
- Reglas de descubrimiento para inserción automática de equipos a monitorizar.
- Permite la monitorización de Logs de sistema y aplicaciones Web.

- El usuario puede definir sus propios parámetros de monitorización (*user parameters*).
- Grandes prestaciones para monitorización de sistemas Windows.

- **A mejorar:**
  - La instalación inicial es un tanto complicada.
  - Se necesita de mucha interacción por parte del usuario en forma de navegación por los distintos menús.
  - La comprensión inicial de los conceptos del funcionamiento de la herramienta exige un cierto tiempo.
  - La interfaz web tiene demasiada densidad de funcionalidades, causando confusión en la navegación para usuarios poco frecuentes.
  - Herramientas como Pandora FMS pueden detectar por sí mismas el tipo de equipo que se está monitorizando y asignar automáticamente una serie de parámetros de monitorización pre-programados para ese tipo concreto. Con Zabbix no existe esa funcionalidad.
  - Actualmente, no es tan conocida como otras herramientas más extendidas (Nagios).
  - Es susceptible de provocar falsas alarmas al no disponer de una funcionalidad similar a los *soft states/hard states* de Nagios.
  - La base de datos utilizada para el almacenamiento necesita de configuración y ajustes.
  - Su uso no está tan extendido como el de otras herramientas.

## ¿Por qué Zabbix?

Como se puede observar en el listado anterior, Zabbix presenta numerosas características a su favor aunque también es cierto que hay aspectos en los que puede mejorar. La necesidad principal que justifica la elección de Zabbix es que, por el momento, de las 3 herramientas estudiadas, es la que mejor soporte ofrece para la monitorización de equipos con sistema operativo Windows. Se considera un motivo de peso porque los equipos más importantes de la infraestructura de CCTV en la UC3M (servidores de grabación) cuentan con ese sistema operativo instalado.

A pesar de no ofrecer mayores prestaciones a nivel de monitorización Windows, Nagios es una herramienta algo más extendida que Zabbix, lo que significa que existe una mayor comunidad de usuarios aportando nuevas ideas y ayudando a solucionar los problemas que puedan registrarse en la herramienta. No obstante, Zabbix cada día cuenta con más adeptos y es posible encontrar mucha información en sus foros oficiales **[39]** e incluso existe un blog de Zabbix en español **[40]** y libros dedicados a esta herramienta. Pandora FMS, por su parte, al ser una herramienta de reciente creación,

aún no ha alcanzado el mismo nivel de usuarios que Zabbix, por lo que no es tan común encontrar información al margen de la documentación de su sitio oficial **[29]**.

Uno de los aspectos más valorados y que motiva, en parte, el haber elegido Zabbix, es que proporciona al usuario una GUI desde la que es posible configurar la práctica totalidad del funcionamiento. No es necesario editar ficheros de configuración (a excepción de los ficheros de los agentes software y el servidor); cualquier parámetro de monitorización, cualquier alerta, cualquier notificación, se puede crear y modificar a través del frontend web. Nagios, por su parte, exige que la mayoría de configuraciones se realicen a través de ficheros de texto, lo cual, además de engorroso, puede conducir a errores. Pandora FMS es muy similar a Zabbix en este contexto de configuración, aunque, tal como indicamos en la tabla anterior, hay ciertos parámetros que deben configurarse en ficheros (es posible hacerlo también a través de la interfaz web de Pandora FMS, pero en su versión de pago).

También destacamos de Zabbix la posibilidad de crear gráficos con los que apoyar la presentación de los datos. En muchos casos el comportamiento del sistema se hace mucho más comprensible a través de representaciones gráficas que por medio de datos numéricos "en crudo". Para cada parámetro de monitorización que tenga un valor numérico, Zabbix puede crear automáticamente gráficos en los que observar la evolución del parámetro en el tiempo. Junto a esos gráficos por defecto, el usuario puede crear sus propios gráficos personalizados para mostrar los valores de monitorización que estime oportunos y, además, por si fuera poco, es posible la creación de pantallas o *screens* que combinen, en un mismo área, la visualización de varios gráficos creados por el usuario. Nagios no es especialmente conocido por su facilidad para la creación de gráficos y Pandora FMS mejora en este aspecto a Nagios, aunque no llega al nivel de Zabbix.

Dentro de los aspectos a mejorar de Zabbix hay una funcionalidad que sería muy deseable encontrar, y que ya hemos tratado en el ANEXO XIII (*soft states/hard states*). Actualmente, y debido a cuestiones relativas a la infraestrucutra de red, hay ocasiones en las que las cámaras de videovigilancia no responden a una petición de icmpping en los tiempos estipulados. Eso supone que se genera un evento con el correspondiente envío de correo electrónico informando de ello y, en la mayoría de los casos, para cuando el administrador del sistema lee ese correo electrónico, la cámara ya está respondiendo de nuevo. Este tipo de falsas alarmas se subsanarían si Zabbix contara con la implementación de la idea de *soft states/hard states*.

Como segundo aspecto que mejoraríamos es el tema del almacenamiento en una base de datos. Si el número de parámetros monitorizados crece y su intervalo de actualización es muy reducido, es posible que, de no haber ajustado convenientemente el rendimiento de la base de datos, nos encontremos con un cuello de botella que resultará en muchos datos de monitorización encolados. Esto es algo de lo que no hay que preocuparse en Nagios.

Por otra parte, cabe señalar que, al igual que la mayoría del software existente, periódicamente se lanzan parches y actualizaciones con los que depurar posibles fallos detectados en la herramienta y mejorar las funcionalidades ya existentes. Los intervalos entre revisiones suelen ser reducidos (sirva como ejemplo que, desde que este proyecto comenzó en septiembre 2009 hasta hoy, en el mes de septiembre de 2010, se han lanzado hasta un total de 7 versiones estables de Zabbix) y, en cualquier caso, lejos de infundir la sensación de que es un software inconsistente, la idea general es que está en constante evolución, algo a lo que contribuyen las sugerencias aportadas por los usuarios.

Tras varios meses de uso, Zabbix se presenta como una de las herramientas más versátiles y completas que podemos encontrar en el ámbito de monitorización de sistemas. En Zabbix encontramos la flexibilidad que necesitamos para poder personalizar la monitorización de los elementos del sistema de videovigilancia en función de su categoría. Así, es posible definir plantillas separadas para los servidores de grabación, los equipos de los centros de control, para las cámaras y para cada tipo de dispositivo concreto de la electrónica de red.

Tal como indicamos en los aspectos destacables de Zabbix, una de las características que valoramos especialmente es el soporte a SNMP, algo que encontramos esencial a la hora de monitorizar el estado de esos equipos en los que no podemos obtener información a través de un agente Zabbix. A excepción de los servidores y los equipos de los centros de control, los demás equipos se monitorizan por medio de información recogida a través de SNMP. No obstante, en aquellos equipos en los que es posible instalar el agente Zabbix, el protocolo SNMP nos sirve para complementar toda esa información obtenida por dicho agente.

Por todo ello y por considerar que Zabbix satisface los requisitos planteados en el apartado 4.1.5., fue escogida como la herramienta en la que basar la plataforma de monitorización del sistema de videovigilancia.

## 5.4. Arquitectura de la plataforma

Una vez tomada la decisión de adoptar Zabbix como herramienta de monitorización, se definió el diseño de la arquitectura de la plataforma de monitorización.

Básicamente, las especificaciones con que cuenta la plataforma son las siguientes:

- Servidor central encargado de recibir toda la información de los equipos monitorizados. El tratamiento de esta información se resume en mostrarla en

la interfaz web de Zabbix, evaluarla comprobando su valor con los umbrales de alerta definidos y, finalmente, almacenarla en la base de datos.

- Agentes Zabbix en los equipos que cuenten con sistema operativo Windows o con sistema operativo Linux. En todos ellos es necesario igualmente tener instalado el agente SNMP para aprovechar las funcionalidades de Zabbix sobre ese protocolo.
- La comunicación entre cada agente Zabbix y el servidor central se realiza a través de los puertos **10050** y **10051** (ambos para TCP). El puerto 10050 es el puerto en el que el agente escucha para la petición de datos por parte del servidor, mientras que el servidor escuchará en su puerto 10051 para el envío de los datos que los agentes envíen periódicamente (*active checks*). Existen dos comportamientos posibles en lo que a la transmisión de datos se refiere. En el primero, que es el más común, el servidor envía peticiones de datos a los agentes con una periodicidad establecida para cada parámetro de monitorización concreta (en ese caso, configuraríamos el parámetro como "*Zabbix agent*") y en el segundo, es el propio agente el que envía datos al servidor sin necesidad de que éste le haga una petición al agente (la configuración para el parámetro sería de tipo "*Zabbix agent (active)*").
- Para comunicarse con los equipos en los que la información se obtenga vía SNMP, se utiliza el puerto ***161*** (UDP).
- El sistema operativo del servidor central es Linux (Ubuntu Server) y la base de datos que incorpora es MySQL.
- La interfaz Web de Zabbix está implementada en PHP y para mostrarse es necesario el servidor Apache como contenedor Web.

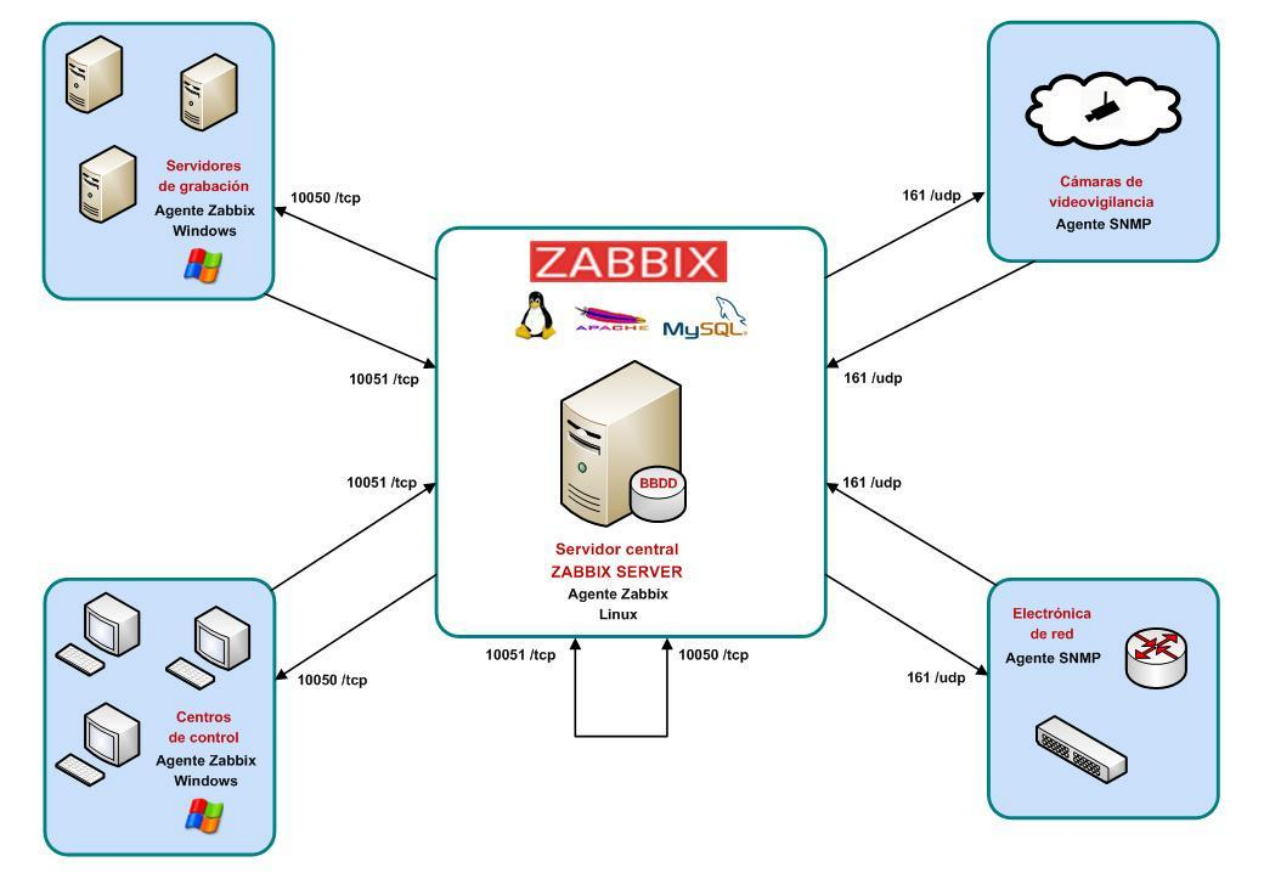



**Figura 15. Arquitectura de la plataforma de monitorización**

## 5.5. Elementos de la plataforma de monitorización

En esta sección describiremos los elementos del núcleo central (servidor central y base de datos) de la Figura 15.

### Servidor central

El servidor central de Zabbix está instalado en la máquina "**zabbix-cctv**". Esta máquina dispone de dos interfaces de red, una de ellas conectada a la red privada de CCTV y la segunda, a la red de la UC3M. Con el fin de evitar que el tráfico procedente de la red '*pública*' UC3M pueda alcanzar la red privada CCTV utilizando esta máquina como *gateway*, se implementan las siguientes medidas de seguridad:

- Se deshabilita la capacidad de enrutamiento entre las dos interfaces desactivando la opción de *forwarding*.

- Control de acceso desde determinadas direcciones IP de la red UC3M a la interfaz web de la plataforma mediante la configuración de un firewall (*iptables*).

El proceso que se encarga de manejar las operaciones relacionadas con la toma de datos de monitorización a través de la interacción con los equipos de la infraestructura de CCTV es "***zabbix_server***" y está ejecutándose de manera constante en el servidor.

El propio servidor tiene instalado un agente Zabbix para monitorizarse a sí mismo, por lo que, además del proceso antes mencionado, también se está ejecutando el proceso "***zabbix_agentd***" en todo momento. Este proceso es el encargado de aceptar las peticiones de datos de monitorización enviadas por el proceso *zabbix_server*.

Además del sistema operativo, se debe instalar todo el software que requiere la herramienta Zabbix para su correcto funcionamiento. *Apache,* módulos *PHP* para mostrar la interfaz Web de la herramienta, y *MySQL*, para gestionar la base de datos de almacenamiento de los datos de monitorización, se encuentran instalados en nuestro servidor central.

En cuanto a la comunicación del servidor con el resto de elementos de la plataforma, tal como se aprecia en la Figura 15, las conexiones se realizan a través del agente instalado en el equipo o bien, en caso de no poder instalarlo, a través del protocolo SNMP.

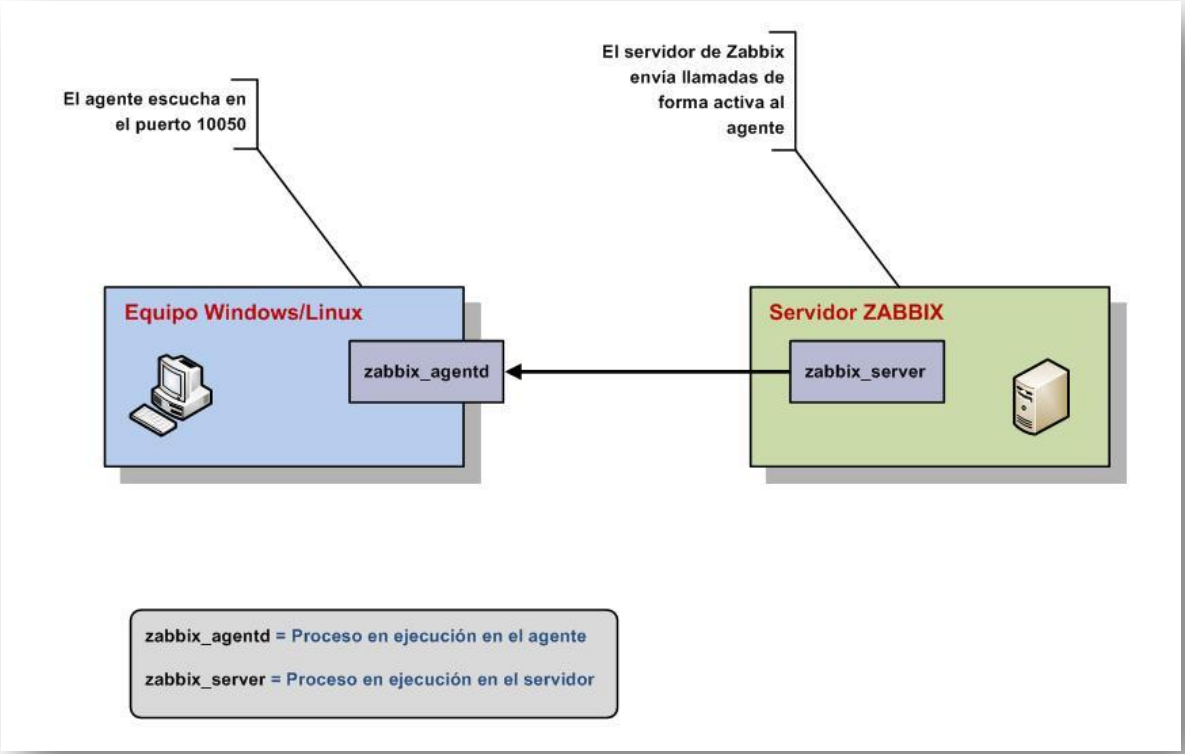



**Figura 16. Comunicación entre el servidor y el agente Zabbix**

Para la monitorización de los servidores de grabación (Windows), los equipos de los centros de control (Windows) y el propio servidor (Linux), el comportamiento es el descrito por la Figura 16. El servidor envía **activamente** peticiones al puerto 10050 del equipo en el que está instalado el agente para así obtener los datos de monitorización.

Existe una variante de este comportamiento en la que es el propio agente quien envía los datos de monitorización sin que haya una petición en su puerto 10050 TCP desde el servidor. En este modo de funcionamiento, el servidor escucha **pasivamente** conexiones en su puerto 10051 TCP a través de las cuales llegan esos datos de monitorización enviados por los agentes. En la siguiente página, la Figura 17 describe este modo de comunicación.

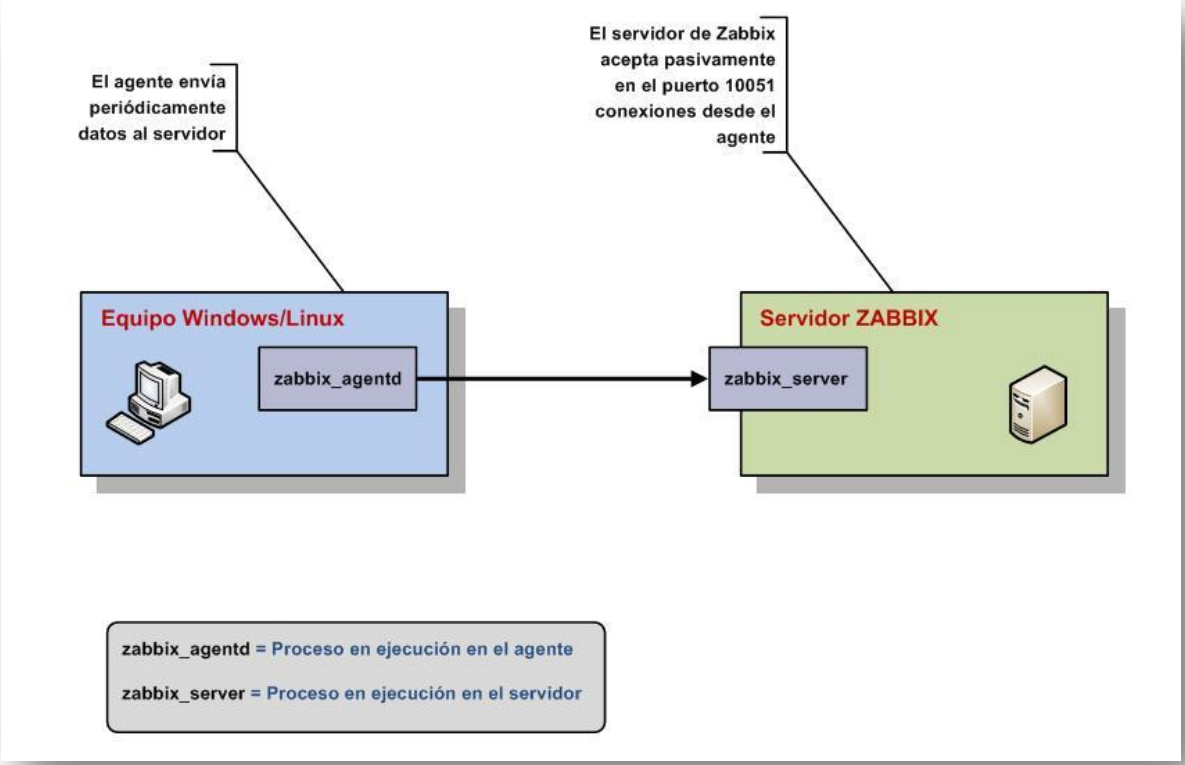



**Figura 17. Comunicación entre el servidor y el agente Zabbix (activa)**

En el resto de elementos que no disponen de un sistema operativo de usuario instalado y, por tanto, no pueden tener un agente Zabbix, la monitorización se realiza a través de SNMP, tal como vemos en el caso de las cámaras de videovigilancia y los equipos que forman la electrónica de red (conmutadores centrales de campus, etc.). Para los equipos Windows y el servidor central también está disponible la monitorización a través de SNMP adicionalmente sobre la monitorización a través del agente Zabbix.

Para esos equipos sin agente Zabbix, los datos de monitorización se obtienen a través del agente SNMP mediante conexiones al puerto 161 de UDP, tal como se aprecia en la Figura 18 a continuación.

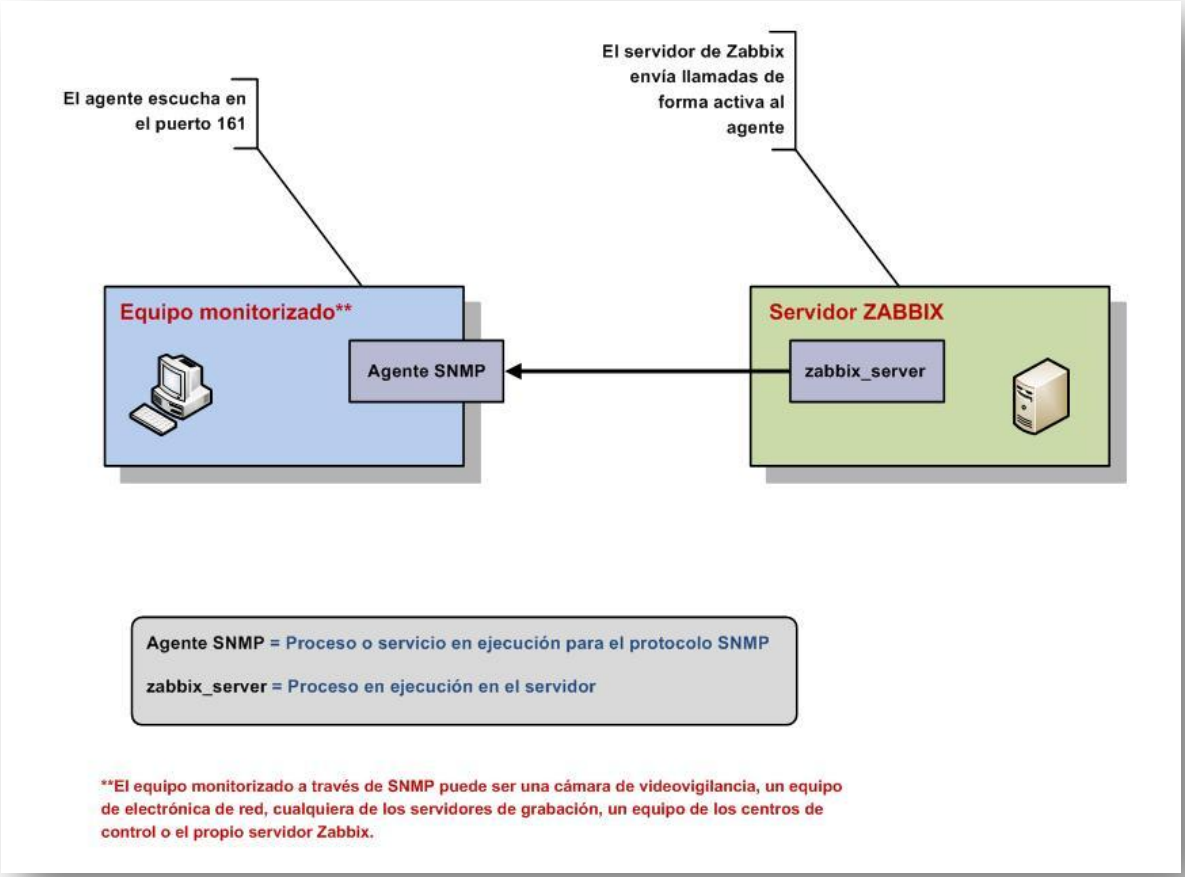



**Figura 18. Comunicación entre el servidor y el agente SNMP**

## Base de datos

El número de equipos monitorizados en la plataforma se resume como sigue:

- 12 servidores de grabación
- 4 equipos en los centros de control
- 246 cámaras de videovigilancia
- 31 dispositivos de electrónica de red
- Servidor central Zabbix

Entre todos ellos, suman **294 elementos**. Por tanto, y a la vista de los requisitos establecidos para la instalación de Zabbix (ver Tabla 7), se decidió que el motor para la base de datos en la que se guardan los datos recogidos fuese *MySQL*. Por otra parte, se trata de una base de datos con la que es fácil trabajar dada la documentación existente.

En esa base de datos se almacenarán no sólo los datos extraídos del proceso de monitorización, sino también la configuración que se haga sobre la herramienta a través de su interfaz Web.

## 5.6. Estructura de los parámetros de monitorización

Los datos obtenidos en el proceso de monitorización del sistema corresponden a lo que Zabbix denomina *items*. Un *item* no es más que un elemento o parámetro de monitorización sobre un equipo, como puede ser el espacio en disco, el tráfico de entrada o cualquiera de las entradas que podemos encontrar en el apartado Necesidades para cada tipo de equipo del presente documento.

En Zabbix hay dos maneras de asignar esos *items* a un equipo. Lo podemos hacer de manera que asignemos uno a uno los items que queramos o bien lo podemos hacer en forma de plantillas o *templates*. Un *template* es una agrupación de items que se puede asignar a un equipo. Esto nos facilita la labor de actualización de los parámetros de monitorización. Para un grupo homogéneo de equipos que comparten las mismas características, en lugar de ir asignando item a item a cada equipo, crearemos una única plantilla en la que estén todos esos items. Al asignar esa plantilla a dichos equipos, estaremos asignándoles también todos los items incluidos en ella. Con esto reducimos el esfuerzo en el caso de que tengamos un elevado número de parámetros a monitorizar.

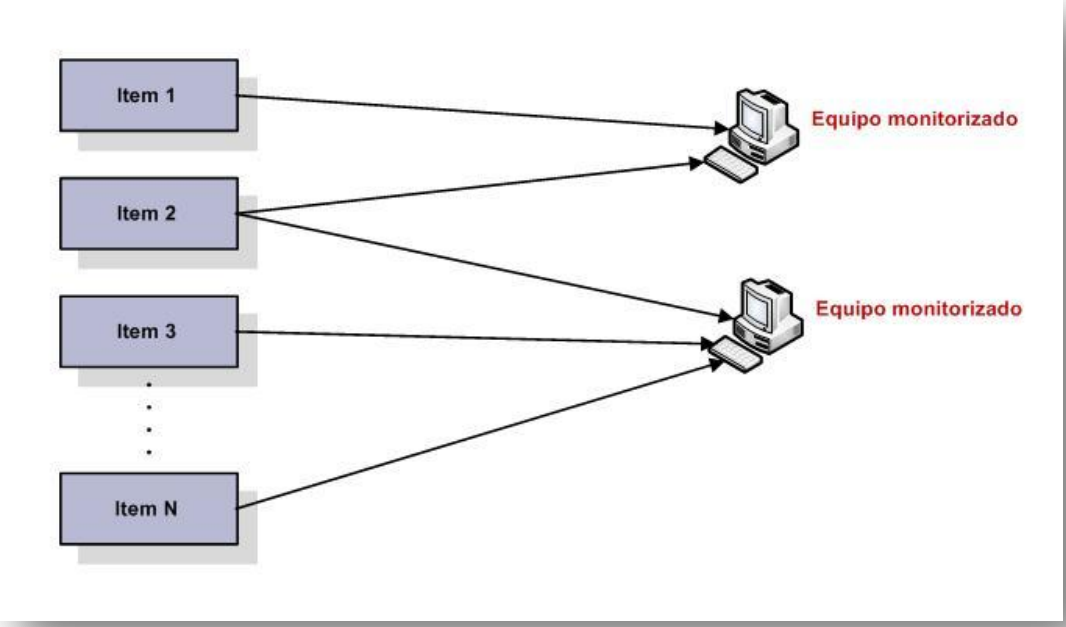



**Figura 19. Estructura de monitorización basada en asignación de items**

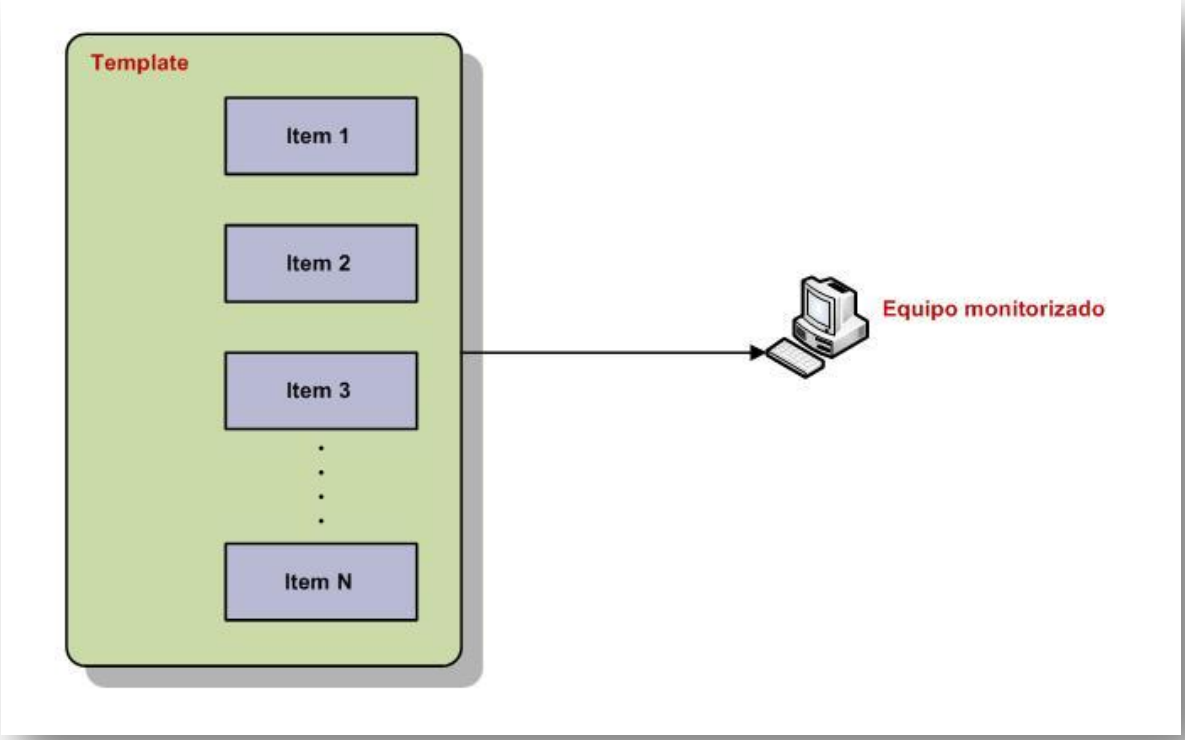



**Figura 20. Estructura de monitorización basada en asignación de plantillas**

## 5.7. Descripción del funcionamiento

Ya hemos visto cómo funciona la comunicación entre el servidor y los equipos de la plataforma para lograr obtener datos de monitorización. Ahora describiremos el modo en que la herramienta de monitorización actúa una vez se reciben esos datos en el servidor central.

Conforme se van recopilando datos, el servidor compara cada uno de esos valores con los umbrales de alerta que se hayan definido previamente. Si el valor recibido sobrepasa el umbral de lo que se considera como "normal" o si, simplemente, corresponde con el valor que hace saltar la alerta, el sistema avisará convenientemente a través de los medios posibles (interfaz Web, e-mail) y, si se han definido, ejecutarán las acciones correspondientes asociadas al evento generado.

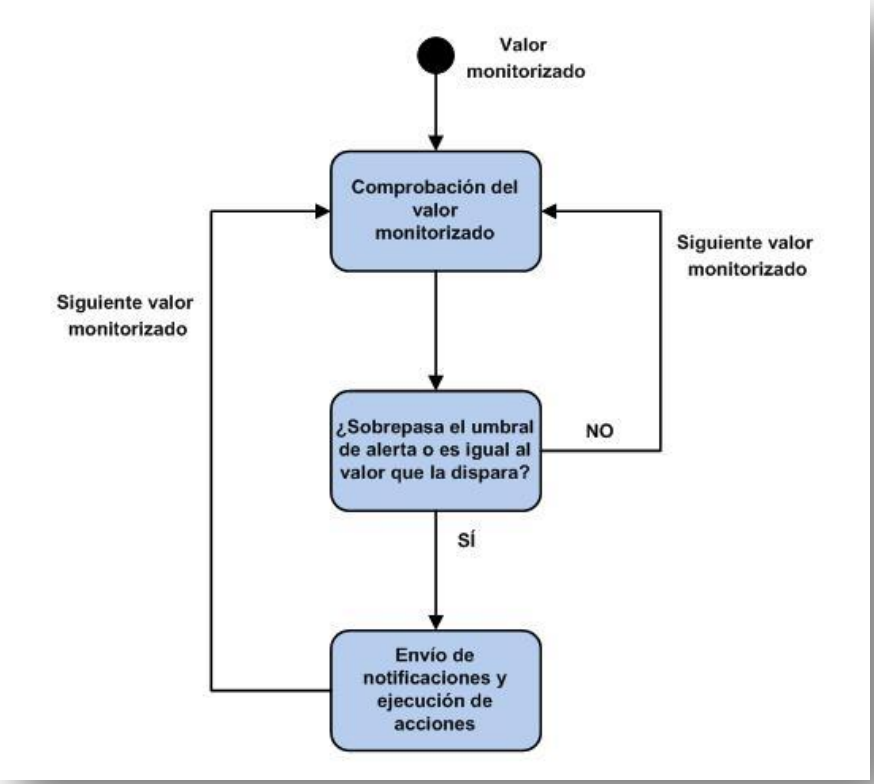



**Figura 21. Diagrama de actuación de las alertas de la plataforma**

Para las alertas se definen unos niveles de criticidad en base a los cuales se definen las notificaciones a enviar y acciones a ejecutar:

- **Sin clasificar (*Not classified*)**
  - Estas alertas se notificarán únicamente a través de la interfaz Web y no llevan asociada ninguna acción. No se ha definido ninguna alerta de este tipo para la plataforma de monitorización actual.

- **Información (*Information*)**
  - Se notificarán a través de la interfaz Web y, dependiendo del tipo de alerta concreta, también se ejecutará una acción de envío de e-mail.

    Las alertas de este tipo que se han diseñado son:

    - Aviso de reinicio en el servidor de Zabbix.
    - Cambios en la configuración del host en el servidor de Zabbix.
    - Cambio del nombre del servidor de Zabbix.
    - Comienzo y finalización de las operaciones de verificación y reconstrucción de los sistemas RAID de almacenamiento de los servidores de grabación.

- Eventos de información registrados en los Logs de Seguridad, Sistema y Aplicación correspondientes a los servidores de grabación y los equipos de los centros de control.

- **Advertencia (*Warning*)**
  - Los criterios son los mismos que para el caso de las alertas de nivel *Información.*

    Las alertas diseñadas para este nivel son:

    - Enlace caído en un puerto de los conmutadores de la electrónica de red.
    - Equipo sin conexión (electrónica de red).
    - Carga de CPU excesiva durante los últimos 3 minutos en el servidor central Zabbix.
    - Alta carga de procesador en el servidor central Zabbix.
    - Excedido el 90% de uso de CPU en los equipos del centro de control o en los servidores de grabación.
    - Eventos de advertencia registrados en los Logs de Seguridad, Sistema y Aplicación correspondientes a los equipos de los centros de control o a los servidores de grabación.
    - Cambio de versión del agente Zabbix instalado en los equipos de los centros de control, servidores de grabación o el servidor de Zabbix.

- **Medio (*Average*)**
  - La forma de actuar es la misma que para las alertas con nivel *Información* y *Advertencia*.

    Las alertas diseñadas para este nivel son:

    - Los demonios *syslogd, sshd, inetd y mysqld* no están ejecutándose en el servidor central Zabbix.
    - Demasiados usuarios conectados al servidor central Zabbix.
    - Falta de memoria libre en el servidor central Zabbix.
    - Cambios en la configuración de los ficheros */usr/sbin/sshd, /usr/bin/ssh, /etc/services, /etc/passwd, /etc/inetd.conf* en el servidor central Zabbix.
    - Demasiados procesos ejecutándose en el servidor central Zabbix.
    - El servidor SSH no está ejecutándose en el servidor central Zabbix.
    - Falta de memoria libre en los equipos de los centros de control o en los servidores de grabación.
    - Reinicio de un servidor de grabación o de un equipo de los centros de control.
    - Cambios en la configuración del fichero "*autoexec.bat*".

- Cambios en la información de los equipos de los centros de control o en los servidores de grabación.
- Alta carga de procesador en los equipos de los centros de control o en los servidores de grabación.
- Uso elevado de CPU en los equipos de los centros de control o en los servidores de grabación.

- **Alto (*High*)**
  - En este caso, la alerta se notificará a través de la interfaz Web y, en todos los casos, se enviará un correo electrónico informando del evento una vez se ha producido y otro correo electrónico una vez se haya solucionado el problema.

    Las alertas diseñadas para el nivel alto son:

    - Cámara de videovigilancia sin conexión.
    - Bajo nivel de espacio en disco en el servidor central Zabbix.
    - El proceso *zabbix_server* no se está ejecutando en el servidor central Zabbix.
    - El proceso *zabbix_agentd* no se está ejecutando en el servidor central Zabbix.
    - El servidor de Apache no se está ejecutando en el servidor central.
    - El servidor HTTP no se está ejecutando en el servidor central.
    - El servidor central Zabbix está caído (no responde a conexión).
    - Demasiados procesos ejecutándose en el servidor central.
    - Demasiados procesos ejecutándose en los equipos de los centros de control o en los servidores de grabación.
    - Fallo en uno de los discos del RAID de almacenamiento de los servidores de grabación.
    - Sistema RAID degradado en uno de los servidores de grabación.
    - Fallo en uno de los 6 ventiladores de uno de los servidores de grabación.
    - Temperatura de la CPU por encima de los 60ºC en uno de los servidores de grabación.
    - Temperatura del sistema por encima de los 60ºC en uno de los servidores de grabación.
    - Bajo nivel de espacio disponible (<10 MB) en disco en los equipos de los centros de control o en los servidores de grabación.
    - Espacio libre en disco por debajo del 5% de la capacidad total en los servidores de grabación.
    - Uno de los servidores de grabación o uno de los equipos de los centros de control está caído (sin conexión).
    - El agente Zabbix no está ejecutándose en un servidor de grabación o en uno de los equipos de los centros de control.

- El software de gestión *Sony RealShot Manager* no está ejecutándose en los servidores de grabación o en los equipos de los centros de control.
- Eventos de error registrados en los Logs de Seguridad, Sistema y Aplicación correspondientes a los equipos de los centros de control o a los servidores de grabación.

- **Crítico (*Disaster*)**
  - Las consideraciones son las mismas que para el caso del nivel *Alto*.

    No se ha diseñado ninguna alerta con este nivel de criticidad.

## 5.8. Interacción con los usuarios

El usuario final de la aplicación es el administrador de la plataforma de monitorización, miembro del área de Seguridad del servicio de informática y comunicaciones de la UC3M. La interacción de este usuario con la plataforma se realizará a través de tres vías o canales:

- **Cliente SSH para conexión remota con el servidor central.**
- **La interfaz Web de la herramienta Zabbix.**
- **Notificaciones de alertas a través del correo electrónico.**

### Cliente SSH

A través del cliente SSH el administrador se conecta desde una terminal al servidor central. Así podrá observar qué procesos se están ejecutando en el sistema (incluyendo el servidor de Zabbix), conectarse a la base de datos para el mantenimiento y administración de la misma y modificar la configuración tanto del servidor Zabbix como del agente instalado en éste.

El acceso por esta vía está restringido a los usuarios que disponen de cuenta de administrador en el servidor central Zabbix.

### Interfaz Web

Tal como se ha mencionado ya en otros apartados, la herramienta Zabbix cuenta con una interfaz Web desde donde los usuarios pueden consultar en cada momento el estado de los equipos monitorizados y de las alertas que se van registrando. Desde la interfaz también se pueden configurar los parámetros de monitorización de forma

personalizada para cada equipo o grupo de equipos y todas las funciones de control y supervisión en la plataforma, como lo son las alertas, los privilegios de acceso, las notificaciones asociadas a alertas, etc.

Para acceder a la interfaz Web se solicita un usuario y contraseña. Inicialmente, se tienen dos tipos de usuario: el **administrador** y un **usuario de monitorización genérico** que cuenta con privilegios sólo para la visualización de datos y no para cambios en la configuración. También es posible la definición de grupos de usuarios. Así, en este caso, se crea un grupo de administradores y un grupo de usuarios de monitorización. Si deseamos añadir más usuarios/grupos de usuarios o simplemente restringir o ampliar los privilegios sobre ellos, lo haremos a través del menú correspondiente de la interfaz.

Desde la interfaz Web se puede acceder a los mapas en los que se muestra el estado de cada equipo. En estos mapas veremos un mensaje "OK" cuando el equipo no presente ningún problema y, en caso de que haya saltado alguna alerta en él, se nos mostrará el problema concreto.

Al igual que los mapas, desde la interfaz también es posible el visualizado de gráficos, ya sean los gráficos que crea automáticamente el servidor como los gráficos definidos por el usuario. En todos ellos podremos observar el rendimiento del servidor a lo largo del tiempo para unos determinados parámetros de monitorización.

### Notificaciones a través del correo electrónico

La notificación del estado de las alertas se puede realizar a través de la propia interfaz Web y por medio de acciones específicas, como puede ser el envío de un correo electrónico a un usuario de la aplicación.

En el caso de la plataforma de monitorización de CCTV, se enviarán correos electrónicos al administrador de la plataforma cada vez que se produzca una alerta de cierto nivel de criticidad. Igualmente, cuando el problema quede solucionado, se enviará un nuevo correo electrónico informando de que dicho problema ha quedado solventado.

La información enviada en esos mensajes será la alerta que registra el evento, el estado de la misma, la dirección IP del equipo en que se ha producido el problema, la hora y fecha en que se registra el evento y el último valor recogido para el parámetro monitorizado que hizo saltar la alerta.

Estos tres canales de interacción quedan reflejados en la siguiente figura:

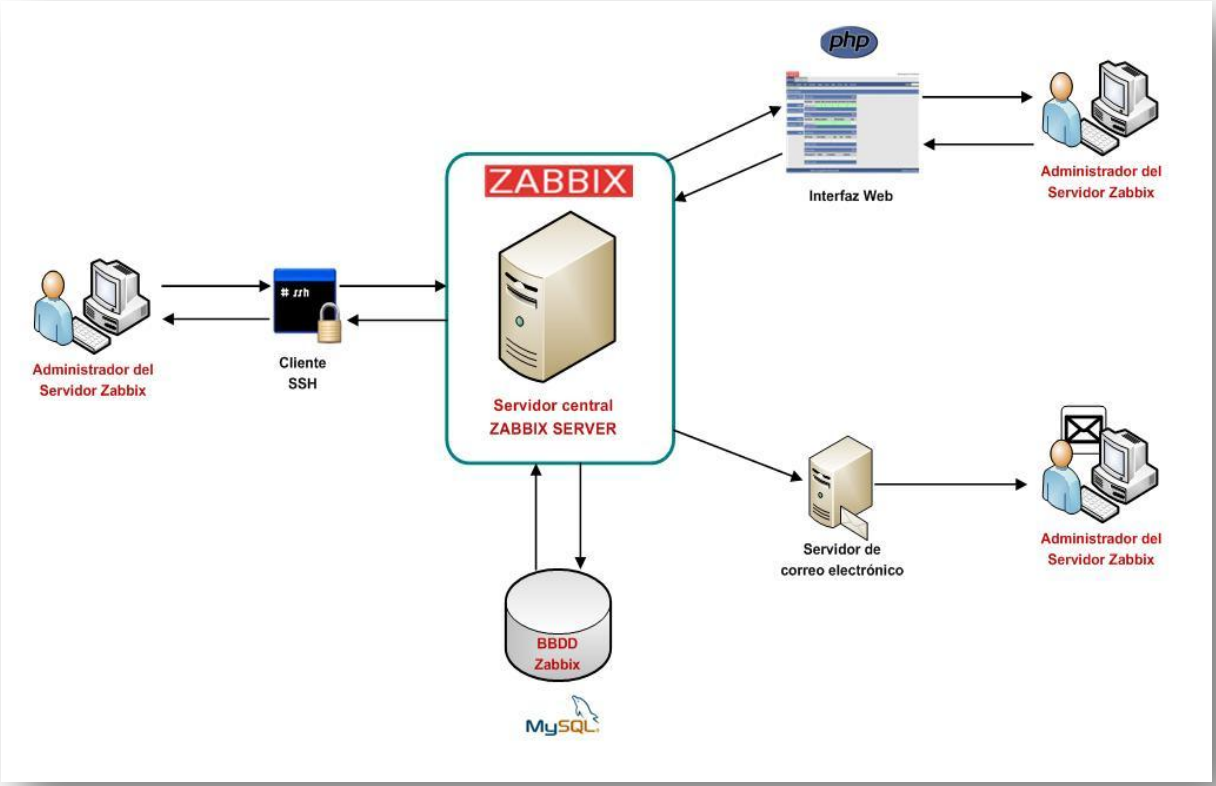



**Figura 22. Diagrama de interacción de la plataforma con los usuarios**

# 6

# Despliegue de la plataforma de monitorización

# 6. DESPLIEGUE DE LA PLATAFORMA DE MONITORIZACIÓN

En el despliegue de la plataforma de monitorización trataremos las principales acciones a llevar a cabo para implementar las soluciones de diseño indicadas en la sección anterior de este documento.

## 6.1. Plan de despliegue

La lista de etapas a seguir en el proceso de despliegue de la plataforma de monitorización es la siguiente:

- Instalación del servidor central en el que se alojará la herramienta Zabbix. Incluye la instalación del sistema operativo y todo el *software* adicional necesario para que Zabbix funcione.
- Instalación y configuración de la herramienta Zabbix en el servidor central.
- Instalación y configuración de los agentes (tanto Windows como Linux) en aquellos equipos de la plataforma en los que sea posible.
- Instalación de los agentes SNMP en los equipos de la plataforma que necesiten de este protocolo para su monitorización.
- Introducción de equipos a monitorizar en la plataforma (servidores de grabación, cámaras, equipos de los centros de control, electrónica de red).
- Establecer los parámetros de monitorización para cada tipo de equipo introducido en la plataforma.
- Creación de alertas.
- Creación de mapas y gráficos.
- Creación de reglas de descubrimiento con las que detectar nuevos equipos conectados a la red CCTV.

## 6.2. Instalación del servidor central Zabbix

El servidor "***zabbix-cctv***", en el que se encuentra instalada la herramienta Zabbix, es un servidor modelo **HP Proliant DL 360 G4** y cuenta con las siguientes especificaciones:

- Procesador Intel Xeon @ 3 Ghz.
- Memoria RAM de 2 GB.
- Discos duros Ultra SCSI de 300 GB (x2).

El primer paso es instalar el sistema operativo en este servidor central. La versión escogida es Linux Ubuntu **[42]** Server en versión 9.04.

Durante la instalación del sistema operativo marcaremos la opción de instalar *LAMP* (Linux + Apache + MySQL + PHP) para así disponer de las aplicaciones necesarias para hacer funcionar la herramienta Zabbix.

Una vez tengamos instalado el sistema operativo y creados los correspondientes usuarios administradores, si deseamos poder arrancar la propia interfaz Web de la que dispone Zabbix, será obligado instalar un entorno gráfico mínimo en el sistema. Para ello, iniciaremos sesión con uno de esos usuarios creados y, desde línea de comandos, escribiremos lo siguiente:

```
$ apt-get install gnome-core
$ apt-get install x-window-system-core
$ apt-get install gnome
```

Llegados a este punto tendremos el sistema operativo instalado con un entorno gráfico mínimo y podremos pasar a la siguiente fase, en la que instalaremos la herramienta Zabbix en nuestro servidor central.

## 6.3. Instalación de la herramienta Zabbix en el servidor central

En este apartado explicaremos la instalación del servidor y el agente Zabbix en el servidor central. La versión instalada actualmente es la versión **1.8.2.** Antes de proceder a la instalación, debemos asegurarnos que nuestro sistema cumple con los requisitos tanto hardware como software establecidos en el apartado de requisitos de Zabbix de la sección correspondiente en Estado de la cuestión.

### Descarga de las fuentes de Zabbix

Si cumplimos todos esos requisitos, continuaremos con la instalación. Hay varios métodos para conseguir las fuentes de Zabbix, pero la más recomendable es recurrir a la última versión estable desde la página oficial de Zabbix **[1]**. Allí, dentro de la sección "Download" descargaremos el paquete comprimido con las fuentes. Normalmente, en la página de descarga encontraremos sólo la última versión estable.

Para facilitar la labor de instalación, creamos un directorio en el que trabajaremos en ello. Por ejemplo, *~/zabbix* (donde ~ es el home del usuario actual). Descargaremos el paquete Zabbix en ese directorio.

## Compilación

Con el archivo comprimido descargado, abriremos un terminal para descomprimirlo:

```
$ cd ~/zabbix; tar –zxvf zabbix-1.8*.tar.gz
```

Compilaremos Zabbix con soporte para el servidor, el agente, MySQL, curl, SNMP e IPMI.

Escribiremos en un terminal:

```
$ cd zabbix-1.8*
$ ./configure --enable-server --with-mysql --with-net-snmp --with-libcurl
--with-openipmi --enable-agent
```

El resumen de la configuración debería mostrarnos un mensaje como el siguiente:

| | |
|---|---|
| **Enable server:** | **yes** |
| **With database:** | **MySQL** |
| **WEB Monitoring via:** | **cURL** |
| **SNMP:** | **net-snmp** |
| **IPMI:** | **openipmi** |
| **Enable agent:** | **yes** |

A continuación compilaremos las fuentes escribiendo:

```
$ make
```

La compilación no toma mucho tiempo, así que después de unos minutos veremos si el proceso de compilación ha terminado con éxito.

El siguiente paso es la instalación propiamente dicha. La manera más habitual es escribir en la línea de comandos la instrucción "*make install*", pero, en lugar de ello, crearemos un paquete de instalación específico para nuestra distribución, y lo haremos con la herramienta ***checkinstall*** **[12].** Entonces, para crear el paquete de Zabbix, instalaremos previamente el software *checkinstall* y escribiremos (como usuario root):

```
# checkinstall --nodoc --install=yes -y
```

Con esto se creará e instalará un paquete que podrá desinstalarse en un futuro a través del gestor de paquetes del sistema.

## Configuración inicial

Tras la compilación, se deben configurar unos parámetros básicos para el servidor y para el agente. Hay ficheros de configuración de ejemplo incluidos en el paquete Zabbix que descargamos previamente. Nuevamente, como usuario *root*, escribimos:

```
# mkdir /etc/zabbix
# cp misc/conf/{zabbix_server.conf,zabbix_agentd.conf} /etc/zabbix
```

Ahora, para configurar el servidor Zabbix haremos unos cambios. Abriremos el archivo */etc/zabbix/zabbix_server.conf* con un editor cualquiera y buscaremos las siguientes entradas en él:

- DBName
- DBUser
- DBPassword

El parámetro "*DBName*" toma el valor "zabbix" por defecto. Para "*DBUser*" escogeremos también el valor "zabbix". En cuanto a "*DBPassword*", escribiremos nuestra contraseña para la base de datos en la que se guardará la información recogida en la monitorización. La contraseña está escrita en claro, así que, para no comprometer su confidencialidad, estableceremos permisos de acceso en el fichero *zabbix_server.conf*:

```
chmod 400 /etc/zabbix/zabbix_server.conf
chown zabbix /etc/zabbix/zabbix_server.conf
```

## Creación de la base de datos

Para que el servidor Zabbix almacene información, necesitamos crear la base de datos. Arrancamos el cliente MySQL:

```
$ mysql –u root -p
```

Introduciremos la contraseña para el usuario *root* (es la contraseña que configuramos en el momento en que instalamos el paquete LAMP).

Una vez hayamos accedido al cliente, podemos crear la base de datos, el usuario con el que Zabbix se conectará a ésta y los permisos necesarios a conceder a este usuario sobre la base de datos creada:

```
mysql> create database zabbix;
  Query OK, 1 row affected (0.01 sec)

mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost'
identified by 'password';
  Query OK, 0 rows affected (0.1 sec)
```

La contraseña con la que identificamos al usuario "*zabbix*" es la misma que escribimos en el fichero de configuración *zabbix_server.conf*.

Ahora cerramos el cliente MySQL:

```
mysql> quit
```

Podemos introducir datos iniciales en la base de datos que acabamos de crear y lo haremos con los ficheros de definición y de datos incluidos en el paquete Zabbix descargado:

```
$ mysql -u zabbix -p zabbix < create/schema/mysql.sql
$ mysql -u zabbix -p zabbix < create/data/data.sql
```

También tenemos la opción de insertar imágenes que usaremos en los mapas de red que crearemos posteriormente. Estas imágenes no son un requisito indispensable para obtener una funcionalidad básica, pero nos serán de ayuda, como decíamos, en la creación de mapas.

```
$ mysql -u zabbix -p zabbix < create/data/images_mysql.sql
```

Estos tres procesos (creación, introducción de datos iniciales, importación de imágenes) deberían completarse de forma normal sin registrarse ningún error (en caso contrario, se deberán revisar los mensajes de error y repetir las operaciones).

Es en este punto cuando tendremos el agente y el servidor Zabbix instalados para comenzar a utilizarlos.

## Comenzando a usar Zabbix

Nunca deberían arrancarse el servidor ni el agente Zabbix con el usuario *root*, así que creamos un usuario bajo el cual ejecutar los procesos correspondientes a los demonios del servidor y el agente Zabbix. Escribiremos, como usuario *root*:

```
# useradd –m –s /bin/bash zabbix
```

Con ello crearemos un usuario llamado *zabbix* cuyo directorio *home* es */home/zabbix* y cuyo intérprete es */bin/bash*.

Para el primer arranque tanto del servidor como del agente Zabbix lo haremos con el usuario *zabbix* recientemente creado:

```
# su – zabbix
$ /usr/local/sbin/zabbix_agentd
```

Así arrancaremos el demonio para el agente Zabbix. Si la ejecución de estos comandos produce error, debemos resolverlos antes de continuar. En caso contrario, podemos continuar con la ejecución del servidor Zabbix (también con el usuario *zabbix*):

```
$ /usr/local/sbin/zabbix_server
```

Si la ejecución manual de tanto el agente como el servidor no ha supuesto problemas, lo siguiente es incluirlos en el arranque del sistema para que así se ejecuten de forma automática con cada inicio del sistema operativo. Esta tarea depende de la distribución que tengamos instalada y en el paquete Zabbix que descargamos en un inicio encontraremos scripts de inicio para determinadas versiones del sistema operativo Linux (SUSE, Slackware, Debian).

## La interfaz Web

Ya tenemos el servidor y el agente Zabbix compilados, instalados y ejecutándose. El siguiente componente a instalar es la interfaz Web a través de la cual observaremos los datos de monitorización que nos proveerá el servidor central. En teoría Zabbix puede contar con múltiples *frontends* aunque sólo uno de ellos tiene funcionalidad completa y éste es el *frontend* o interfaz Web, escrito en PHP.

Existe una serie de prerrequisitos que deben cumplirse para mostrar la interfaz Web. Como toda interfaz, requiere de una plataforma en la cual ejecutarse y ésta es, en este caso, un servidor web con entorno PHP. Necesitaremos instalar:

- Servidor Web soportado por PHP; Apache es la opción comúnmente escogida.
- PHP, al menos con versión 5.

Instalaremos estos dos componentes desde paquetes específicos para nuestra distribución. En el caso de PHP, además, necesitaremos las siguientes funcionalidades:

- php-gd
- php-mysql
- php-bcmath

Una vez instalado todo lo necesario, configuraremos el frontend Web. En primer lugar, debemos decidir dónde alojaremos el código del frontend. La mayoría de las distribuciones utilizan la ruta */srv/www/htdocs*, pero, en nuestro caso, escogeremos */var/www/htdocs*.

Entonces, escribimos (como usuario *root*):

```
# cp -r frontends/php /var/www/htdocs
# mv /var/www/htdocs/php /var/www/htdocs/zabbix
```

Ya podemos abrir nuestro navegador Web, en cuya barra de direcciones escribiremos *http://<nombre_del_servidor_o_direccion_ip/zabbix*. Entonces se lanzará el asistente de configuración, del que podremos obtener más información en el anexo ANEXO I. Configuración del frontend Web de Zabbix.

Inmediatamente después de instalar el frontend Web ya podremos acceder a la interfaz de Zabbix, mostrándose en primer lugar la pantalla de login para introducir nuestro usuario y contraseña.


**Figura 23. Inicio de sesión en la herramienta Zabbix**

Iniciaremos sesión con el usuario administrador que hayamos creado durante la instalación de la herramienta y, como pantalla de bienvenida, tendremos una visión general del estado del sistema.

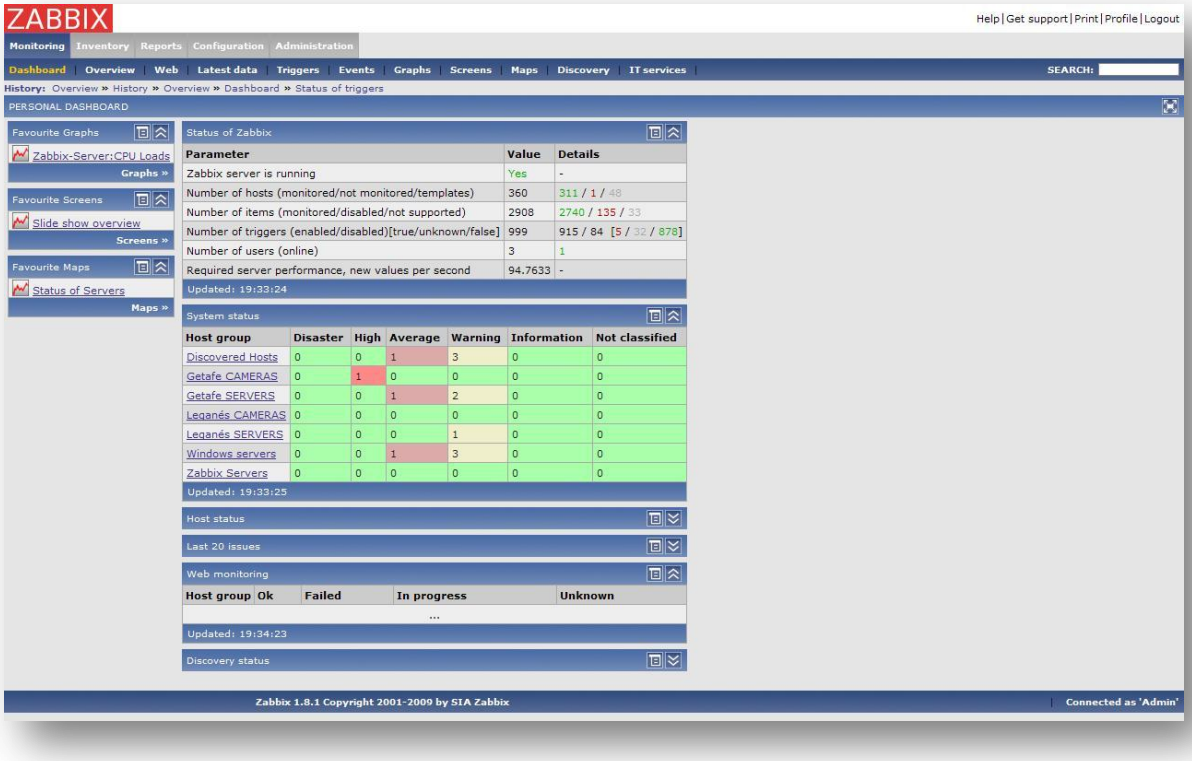



**Figura 24. Estado general del sistema de monitorización**

Como se puede observar, el frontend dispone de un menú navegacional en el que encontramos 5 categorías principales:


**Figura 25. Menú principal del frontend de Zabbix**

- **Monitoring (Monitorización)**: esta categoría contiene enlaces a la mayoría de las páginas relacionadas con la monitorización. Desde aquí se visualizan los datos, los problemas existentes y los gráficos construídos sobre la información recopilada. Los subapartados que nos encontramos en este menú son:
  - **Dashboard (Panel de mando)**: aquí veremos el estado general del sistema (número de equipos monitorizados, nuevos valores por

segundo), el estado de cada equipo de la plataforma, los últimos eventos ocurridos, etc.

- **Overview (Vistazo general)**: desde aquí podremos ver rápidamente los parámetros monitorizados para cada equipo. Dependiendo de si tienen configurada una alerta y, del nivel de ésta, los datos aparecerán resaltados con sombreado de distinto color para identificarlos inmediatamente.
- **Web**: monitorización de las aplicaciones Web. En principio, no se contempla la monitorización de ninguna Web, aunque está en estudio incluir dentro de esta sección la monitorización del propio frontend de Zabbix.
- **Latest data (Últimos datos)**: como su nombre indica, aquí podemos ver los últimos valores de monitorización recogidos. Desde aquí es posible filtrar éstos por nombre así como obtener gráficos con los que observar su evolución.
- **Triggers (Disparadores)**: los disparadores son los mecanismos que hacen que una determinada alarma se active. Aquí podemos ver el estado de esas alarmas (activas o no).
- **Events (Eventos)**: muestran los cambios que se han llevado a cabo producto de la activación de las alertas configuradas. Indica el cambio de estado de la alerta asociada (inactiva a activa y viceversa).
- **Graphs (Gráficos)**: desde aquí podemos ver los gráficos que hayamos creado para cada equipo introducido en la plataforma.
- **Screens (Pantallas)**: las pantallas se construyen partiendo de gráficos ya existentes. Desde este menú podemos visualizarlas.
- **Maps (Mapas)**: aquí veremos los mapas que previamente hayamos creado.
- **Discovery (Descubrimiento)**: se nos muestra aquí el estado de las reglas de descubrimiento que se hayan definido para identificar nuevos equipos conectados a la plataforma.
- **IT services (Servicios IT)**: servicios de alto nivel para visualización del estado de la infraestructura.

- **Inventory (Inventario)**: permite la visualización, si están definidos, de los datos de inventario de los sistemas monitorizados.
  - **Hosts (Equipos de la plataforma)**: si a la hora de introducir un equipo en la plataforma incluimos una dirección URL en su descripción y ésta comienza por *https* o *http*, se creará en esta sección de inventario un enlace a esa dirección.

- **Reports (Informes)**: cada vez que se necesite obtener un informe detallado en el que se visualicen diversos datos en pantalla, esta será la categoría a visitar.

- **Status of Zabbix (Estado de Zabbix)**: aquí podemos ver si el estado de la plataforma en términos del número de equipos monitorizados, número de elementos o parámetros monitorizados, disparadores, eventos generados y alertas lanzadas.
- **Availability Report (Informe de disponibilidad)**: muestra el porcentaje de tiempo durante el cual las alertas han estado activas o inactivas informando así sobre la disponibilidad de cada equipo monitorizado.
- **Most busy triggers top 100 (listado de los 100 disparadores con más actividad)**: se mostrará una lista con el nombre de los 100 disparadores que han registrado más actividad en la plataforma.
- **Bar reports (Informes de barras)**: los informes de barras permiten observar la información de monitorización en mayor detalle que los gráficos definidos por defecto en Zabbix. Desde aquí se pueden crear y personalizar esos informes.

- **Configuration (Configuración)**: constituye el punto desde el cual crear los sistemas monitorizados o hosts, los parámetros a monitorizar en éstos, los mensajes de notificación en caso de alertas, la programación de disparadores, etc.
  - **General**: ofrece funcionalidades como la importación de imágenes y fondos con los que construir mapas, creación de mapeados de valores, configuración del tiempo que está activa la plataforma, etc.
  - **Host groups (Grupos de equipos)**: desde esta sección se podrán crear grupos dentro de los cuales organizar los equipos de la plataforma.
  - **Hosts (Equipos de la plataforma)**: este es el punto más importante, pues desde aquí se introducen los equipos que se vayan a monitorizar en la plataforma.
  - **Maintenance (Mantenimiento)**: permite crear periodos de mantenimiento para la plataforma de monitorización. De esta manera podemos desactivar temporalmente el frontend de Zabbix cuando queramos llevar a cabo operaciones de mantenimiento sobre la base de datos, por poner un ejemplo.
  - **Web**: desde este punto configuraremos las páginas Web que se deseen monitorizar.
  - **Actions (Acciones)**: en este menú podemos crear, borrar, activar y desactivar las acciones (envío de e-mail, etc) asociadas a una determinada alarma. Permite personalizar las condiciones bajo las cuales se ejecutarán las acciones.
  - **Screens (Pantallas)**: a partir de los gráficos creados se podrán componer pantallas en las que mostrar, de un solo vistazo, uno o varios de esos gráficos. También se puede construir una secuencia de diapositivas (slide show) que muestre una secuencia automática de una serie de gráficos o pantallas.

- **Maps (Mapas)**: aquí se pueden crear mapas con los que representar el estado de la plataforma a través de iconos e imágenes.
- **IT services (Servicios IT)**: en esta sección se configuran los servicios de alto nivel para visualización de la infraestructura.
- **Discovery (Descubrimiento)**: desde aquí configuraremos las reglas de descubrimiento para la detección de nuevos equipos conectados a la plataforma.
- **Export/Import (Exportar/Importar)**: son funcionalidades que permiten exportar los equipos y los parámetros monitorizados a un fichero de formato *xml*. Igualmente, la opción de importación permite efectuar la operación inversa. Son útiles para hacer copias de seguridad de la estructura de monitorización de la plataforma.

- **Administración (Administration)**: mientras que la categoría de Configuración se ocupa de los aspectos relacionados con todo aquello que se monitoriza y cómo actuar sobre determinadas condiciones, la categoría de Administración brinda la posibilidad de personalizar los aspectos internos de Zabbix tales como los métodos de autenticación, los usuarios, los permisos y otras tareas similares.
  - **General**: aquí podemos llevar a cabo tareas como la importación de imágenes, creación de mapeados de valores, por citar algunos ejemplos.
  - **DM (Distributed Monitoring – Monitorización distribuida)**: en esta sección se configurarán los nodos y/o proxies de la plataforma de monitorización.
  - **Authentication (Autenticación)**: se puede escoger entre varios métodos de autenticación en la plataforma (*Internal, LDAP, HTTP*), cada uno de ellos con sus respectivas opciones de configuración.
  - **Users (Usuarios)**: aquí añadiremos y configuraremos los usuarios/grupos de usuarios de la herramienta Zabbix. Podemos personalizar los permisos a aplicar a cada uno de esos usuarios o grupo de usuarios.
  - **Media Types (Tipos de medios)**: en este apartado definimos los medios a través de los cuales se comunica Zabbix con los usuarios cuando se produce un evento en el sistema.
  - **Scripts**: desde aquí el usuario puede crear scripts a ejecutar del lado del servidor Zabbix sobre los equipos de la plataforma.
  - **Audit (Auditoría)**: refleja los cambios llevados a cabo en la configuración de Zabbix y los inicios de sesión a través de la interfaz Web.
  - **Queue (Cola de valores)**: muestra los valores de monitorización que están pendientes de ser refrescados. Es simplemente una representación de la información de la base de datos.
  - **Notifications (Notificaciones)**: muestra los mensajes enviados a través de los medios configurados previamente.

- **Locales**: ofrece funcionalidades para edición de traducciones del lenguaje del frontend de Zabbix.
- **Installation (Instalación)**: en este apartado comprobaremos que los componentes software que necesita Zabbix (PHP, base de datos MySQL) funcionan correctamente.

## 6.4. Instalación de los agentes Zabbix

Describimos aquí los pasos seguidos para la instalación de los agentes Zabbix con los que se comunica el servidor Zabbix recientemente instalado.

### 6.4.1. Instalación en Linux

Tenemos varias opciones para instalar el agente. La primera de ellas, a través de repositorios, para lo cual simplemente escribiremos:

```
$ apt-get install zabbix-agent
```

La segunda opción es instalarlo desde las fuentes disponibles en el site de Zabbix. Como primer paso, descargaremos el fichero comprimido con la última versión del agente Zabbix para sistemas Linux desde la página oficial [1], lo descomprimiremos y, a continuación, escribimos:

```
$ ./configure –enable-agent
```

De esta manera comprobaremos si disponemos de todas las librerías y aplicaciones necesarias para que el agente funcione correctamente y lo compilaremos para después proceder a su instalación.

```
$ make install
```

Con esta última instrucción instalaremos el agente.

Independientemente de la opción escogida para la instalación del agente, existe un fichero de configuración del mismo que deberemos editar en función de nuestras necesidades.

```
$ vim /etc/zabbix/zabbix_agentd.conf
```

Se abrirá el fichero de configuración y, si deseamos habilitar la ejecución de comandos remotos en el equipo en el que hemos instalado el agente, desharemos el comentario de la línea correspondiente a los *RemoteCommands*:

```
# Enable remote commands for ZABBIX agent. By default remote commands disabled

EnableRemoteCommands=1
```

Así funcionarían los comandos remotos que se ejecuten del lado de la máquina en la que esté instalado el agente.

El siguiente parámetro de configuración a editar es la dirección IP del servidor central Zabbix.

```
# List of comma delimited IP addresses (or hostname) of ZABBIX servers.
# No spaces allowed. First entry is used for sending active checks.
# Note that hostname must resolve hostname-> IP address and IP address-> hostname

Server=127.0.0.1
```

Por defecto, el puerto en el que el agente escucha al servidor es el 10050. Mantendremos este valor para el agente instalado.

```
# List port. Default is 10050

#ListenPort=10050
```

Podemos definir un nivel de *log* para la depuración del agente. Existen 5 niveles, dentro de los cuales escogeremos el nivel 3 para que nos muestre únicamente los avisos registrados durante la ejecución del agente.

```
# Specifies debug level
# 0 – debug is not created
# 1 – critical information
# 2 – error information
# 3 – warnings
# 4 – information (default)
# 5 – for debugging (produces lots of information)

DebugLevel=3
```

Los mensajes de log que se generen con el nivel que hayamos definido se escriben en un fichero cuya ubicación configuraremos en *LogFile*.

```
# Name of log file
# If not set, syslog will be used

LogFile=/var/log/zabbix-agent/zabbix_agentd.log
```

### 6.4.2. Instalación en Windows

La versión del agente Zabbix para sistemas Windows se instalará en los servidores de grabación y en los equipos de los centros de control. Accederemos a la página oficial de Zabbix y allí descargaremos la última versión del agente para sistemas operativos Windows. En el archivo comprimido que descargamos encontraremos dos carpetas, una con la versión del agente para sistemas de 32 bits y otra para sistemas de 64 bits. Escogemos la versión de 32 bits tanto para los servidores de grabación como para los equipos de los centros de control.

En la carpeta del agente encontraremos, entre otros, tres archivos que copiaremos a una carpeta en la raíz de C: (C:\zabbix):

- **zabbix_agentd.exe**: agente Zabbix propiamente dicho.
- **zabbix_get.exe**: para comprobar el funcionamiento del agente.
- **zabbix_sender.exe**: para enviar paquetes de información al servidor Zabbix.

Además de estos tres archivos, veremos que hay un fichero de configuración "zabbix_agentd.conf" que deberemos editar previamente a la instalación del agente.

En primer lugar, editamos el fichero indicándole la dirección IP del servidor central Zabbix:

```
# List of comma delimited IP addresses (or hostname) of ZABBIX servers.
# No spaces allowed. First entry is used for sending active checks.
# Note that hostname must resolve hostname-> IP address and IP address-> hostname

Server=XXX.XXX.X.X (escribir aquí la dirección IP del servidor Zabbix)
```

Mantenemos el puerto 10050 como puerto por defecto para escuchar las peticiones desde el servidor central y el puerto 10051 para el envío de información desde el agente hacia el servidor (*active checks*).

```
# Server port for sending active checks

ServerPort=10051

# Listen port. Default is 10050

ListenPort=10050
```

Al igual que hicimos en el caso de los agentes instalados en Linux, deshacemos el comentario de la línea reservada para los comandos remotos y con ello habilitarlos:

```
# Enable remote commands for ZABBIX agent. By default remote commands disabled

EnableRemoteCommands=1
```

El nivel de *log* que seleccionaremos es 3.

```
# Specifies debug level
# 0 - no debug
# 1 - critical information
# 2 - error information
# 3 - warnings
# 4 - for debugging (produces lots of information)
#
# Mandatory: no
# Default:

DebugLevel=3
```

Cambiaremos la ubicación del fichero de log en la línea correspondiente del fichero de configuración.

```
# Name of log file.
#
# Mandatory: no
# Default:
# LogFile=

LogFile=C:\zabbix\\Zabbix_agentd.log
```

Por último, cambiaremos el parámetro correspondiente al nombre del equipo en el que hemos instalado el agente. Si no aceptamos el parámetro por defecto (*system.uname*) y escribimos un nombre personalizado, tendremos que asegurarnos de que, en el momento en que introduzcamos este equipo en la plataforma de monitorización, lo hagamos con exactamente el mismo nombre que estamos escribiendo en este fichero de configuración.

```
# Unique hostname.
# Required for active checks and must match hostname as configured on the server.
#
#  Default:
# Hostname=system.uname

Hostname=Servidor 1
```

Una vez se han hecho todos los cambios necesarios en el fichero de configuración, podemos instalar el agente. Escribiremos:

```
zabbix_agentd.exe –i –c zabbix_agentd.conf -s
```

La opción "*-i*" instala el agente, "*-c*" especifica el fichero de configuración que previamente hemos editado y "*-s*" arranca el agente Zabbix como un servicio del sistema.

## 6.5. Instalación SNMP

### 6.5.1. Instalación en Linux

El único paquete que necesitaremos en el servidor es **snmpd**, el demonio de SNMP.

Para instalarlo, escribimos:

```
$ sudo apt-get install snmpd
```

Para la configuración de SNMP existen dos ficheros, ubicados en */etc/default/snmpd* y */etc/snmp/snmpd.conf*, que, en nuestro caso, dejaremos con sus parámetros por defecto.

Si queremos comprobar que la instalación de SNMP funciona, tenemos la utilidad **snmpwalk** [33]. Como ejemplo, podemos lanzar la utilidad para recopilar información sobre uno de los servidores:

```
$ snmpwalk –v 2c –c public 192.168.14.61
```

En esta línea, "-v" indica la versión, que es 2c (versión 2) en este caso, y "-c" se reserva para la comunidad, public en el ejemplo indicado (ver sección 3.3.2 para más información sobre las versiones de SNMP). Finalmente, se incluye la dirección IP del host sobre el cual se lanza la petición SNMP.

### 6.5.2. Instalación en Windows

La instalación de SNMP en los sistemas operativos Windows se realiza desde el *Panel de control* del sistema. Allí, dentro de *Agregar o quitar programas*, iremos a la

opción *Agregar o quitar componentes de Windows* y, una vez allí, seleccionamos la opción *Herramientas de Administración y supervisión*. En este punto seleccionaremos el protocolo SNMP para su instalación, en caso de que no se encontrara instalado. Para más informacion, ver ANEXO IX. Guía de instalación de SNMP en Windows.

Una vez instalado, debemos asegurarnos que se ha instalado un nuevo servicio en el sistema. Para ello, accederemos a la lista de servicios del sistema y comprobaremos que existe una entrada para el servicio SNMP. Para configurarlo, accedemos a las propiedades del servicio y, en la ficha **Seguridad**, indicaremos que se acepten paquetes de cualquier host y estableceremos que "public" sea el único nombre de comunidad aceptado.

### 6.5.3. Activación de SNMP en las cámaras de videovigilancia

El protocolo SNMP debe activarse en las cámaras de videovigilancia según las instrucciones descritas en el manual de las mismas. En ese manual se nos indicará que ejecutemos el comando siguiente:

**http://ip_adr/snmpconf/snmpconf.cgi?<parameter>=<value>&<parameter>=...&...**

Básicamente se trata de indicarle a la cámara a través de un comando CGI una serie de limitaciones de acceso y la comunidad de hosts a los que se permite la lectura de parámetros SNMP en la cámara.

En el caso de las cámaras de videovigilancia del sistema de CCTV de la UC3M el comando establecido es el siguiente:

**http://ip_adr/snmpconf/snmpconf.cgi?community=1,r,public,0.0.0.0**

Con este comando permitiremos que cualquier host de la comunidad "*public*" pueda obtener información vía SNMP de la cámara que tenga la dirección IP dada por "*ip_adr*".

## 6.6. Introducción de equipos en la plataforma

El primer paso para monitorizar el estado de los equipos del sistema de CCTV a través de Zabbix es introducirlos en la plataforma. Ya hemos visto que son dos las formas en que pueden registrarse esos equipos en Zabbix, una de forma manual y otra, a través de reglas de descubrimiento.

Inicialmente, introduciremos los equipos de forma manual y los organizaremos en cuatro grandes grupos correspondientes, respectivamente, a los servidores de grabación, las cámaras de videovigilancia, la electrónica de red y los equipos de los centros de control. Adicionalmente, crearemos un grupo reservado para el propio servidor central Zabbix y así tenerlo monitorizado.

### 6.6.1. Servidores de grabación

En primer lugar, crearemos en Zabbix tres grupos:

- **Windows servers**: grupo que incluye a todos aquellos equipos o *hosts* que tengan a Windows como sistema operativo (todos los servidores de grabación estarían encuadrados en este grupo).
- **Leganés servers**: servidores de grabación del campus de Leganés.
- **Getafe servers**: servidores de grabación del campus de Getafe:

Crearemos los grupos desde el menú *Configuration -> Host Groups.*

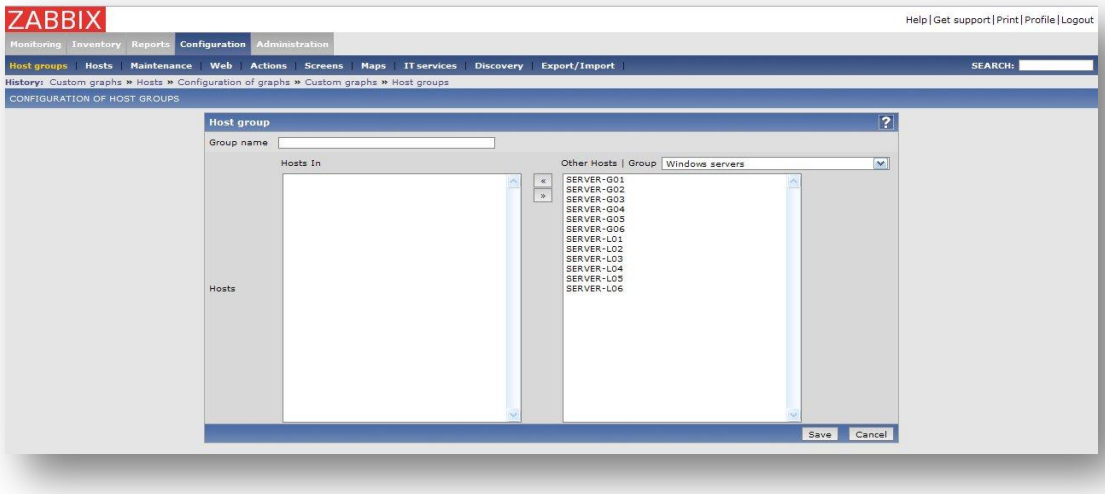



**Figura 26. Creación de grupos de equipos en Zabbix**

Como podemos ver en la figura anterior, a la hora de crear un grupo de equipos o *hosts* podemos asignarle un nombre y una serie de equipos que estarán incluidos en él.

Una vez hayamos creado los tres grupos indicados, es momento de introducir los servidores de grabación. Uno tras otro, iremos insertando los 12 servidores de grabación asignándoles el mismo nombre que escribimos en el fichero de configuración de sus agentes.

En este caso, la utilidad que nos permite introducir los equipos la encontraremos en *Configuration -> Hosts*.

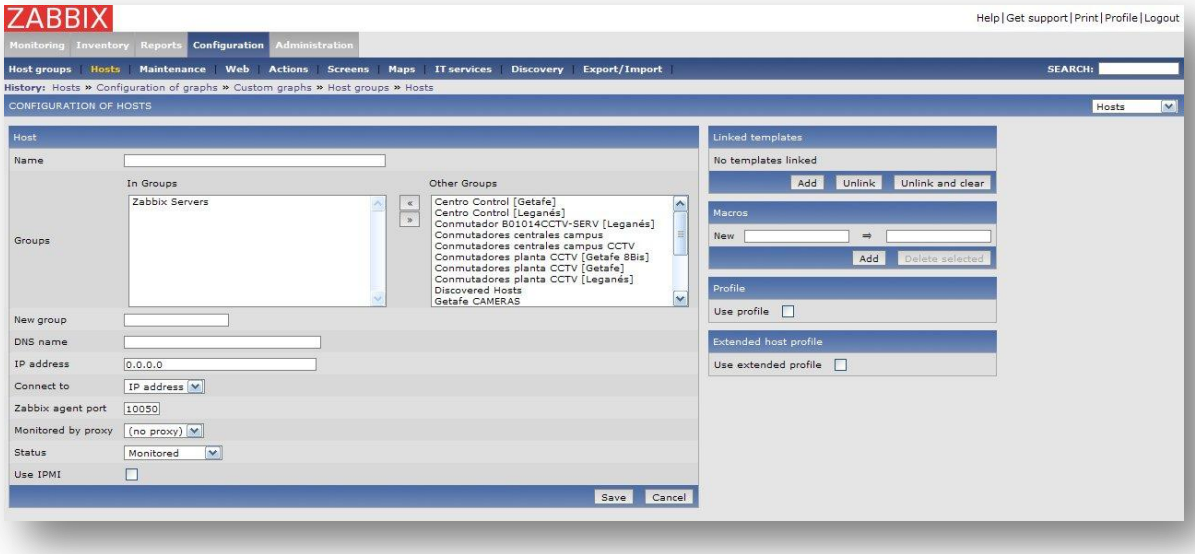



**Figura 27. Creación de equipos en Zabbix**

Para cada equipo podremos establecer un nombre, vincularlo a un grupo o varios grupos, una dirección IP o un nombre DNS con el que identificarlo, el puerto en el que escucha el agente que tenga instalado, un estado (monitorizado o sin monitorizar) y, finalmente, tendremos la opción de enlazar o conectar el equipo con una plantilla en la que están definidos los parámetros que se monitorizarán en el equipo. La creación de estas plantillas se tratará en posteriores apartados.

### 6.6.2. Cámaras de videovigilancia

Para las cámaras de videovigilancia haremos una clasificación similar a los servidores de grabación. Crearemos dos grupos:

- **Getafe cameras**: cámaras del campus de Getafe.
- **Leganés cameras**: cámaras del campus de Leganés.

Introduciremos cada cámara en el grupo que le corresponda desde el mismo menú que antes (*Configuration -> Hosts*).

### 6.6.3. Electrónica de red

Nuevamente, se crean primero grupos diferenciados para cada tipo de equipo de la electrónica de red:

- **Conmutadores de planta [Getafe]:** se incluyen aquí los conmutadores de planta a los que están conectadas las cámaras de videovigilancia en los edificios del campus de Getafe.
- **Conmutadores de planta [Leganés]:** de manera análoga al campus de Getafe, en este grupo estarán incluidos los conmutadores de planta a los que se conectan las cámaras de videovigilancia del campus de Leganés.
- **Conmutadores centrales de campus (red CCTV)**: este grupo engloba a los conmutadores centrales de CCTV vistos en la topología de red descrita en la sección "Análisis del sistema de videovigilancia" (ver Figura 7).
- **Conmutadores centrales de campus (red UC3M)**: se introducen en este grupo los conmutadores a través de los cuales se conectan las subred de CCTV del campus de Getafe con la subred de CCTV del campus de Leganés (ver Figura 7).

### 6.6.4. Equipos de los centros de control

En cada centro de control se monitorizan los dos equipos desde los que se visualizan las imágenes tomadas en tiempo real por las cámaras de videovigilancia. Crearemos dos grupos, uno para el campus de Getafe y otro para el campus de Leganés:

- **Centro Control [Getafe]**: en este grupo se introducirán los equipos del centro de control de Getafe.
- **Centro Control [Leganés]**: aquí se incluirán los equipos del centro de control de Leganés.

### 6.6.5. Servidor central Zabbix

El servidor central Zabbix se introducirá en dos grupos:

- **Zabbix servers**: este grupo se reserva para aquellos equipos que tengan instalado un servidor de monitorización Zabbix.
- **Linux servers**: los equipos aquí incluidos se caracterizan por ser servidores con sistema operativo Linux.

### 6.6.6. Introducción mediante reglas de descubrimiento

La introducción de los equipos en la plataforma se ha llevado a cabo hasta ahora de forma manual a través del menú correspondiente (*Configuration -> Hosts*). Sin embargo, existe otra segunda vía para introducirlos y es a través de **reglas de descubrimiento**.

Ésta es una manera muy útil de introducir los equipos porque ahorra el trabajo que supone hacerlo manualmente. En el momento en que se conecte un equipo dentro del rango de red que nosotros definamos, el equipo se introducirá automáticamente en la plataforma. Las posibilidades que las reglas de descubrimiento de Zabbix ofrecen incluyen la asignación automática de parámetros de monitorización al equipo desde el mismo momento en que éste es introducido.

Para crear una regla de descubrimiento, accedemos al menú *Configuration -> Discovery*. Se nos presentará la siguiente pantalla:

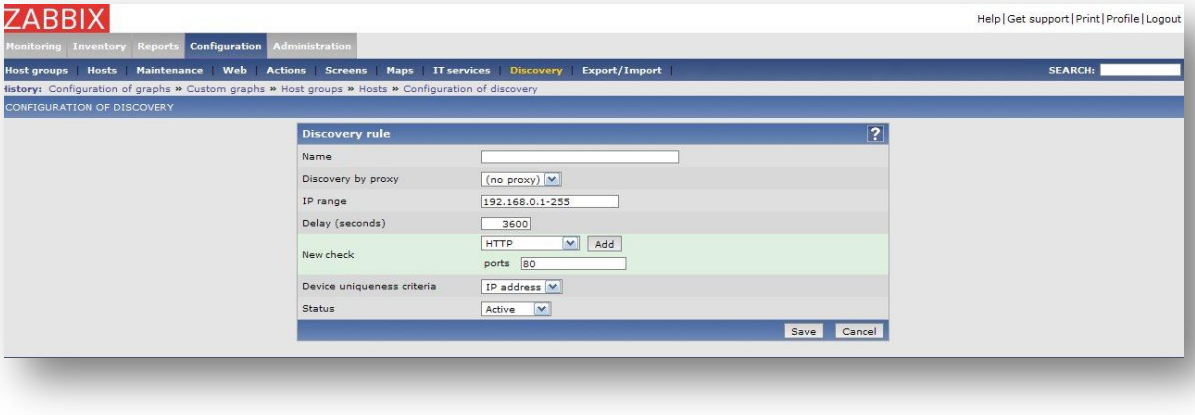



**Figura 28. Creación de reglas de descubrimiento en Zabbix**

Establecemos el rango de direcciones IP dentro del cual se "descubrirían" los nuevos equipos, la frecuencia con la que se ejecutará la regla de descubrimiento (es aconsejable que el intervalo de ejecución sea amplio para así no sobrecargar la CPU) y las condiciones que debe cumplir el equipo para ser considerado como "descubierto" (p.ej. que el equipo tenga activo el agente de Zabbix o el agente SNMP).

Una vez que el nuevo equipo ha sido detectado con las reglas de descubrimiento, deberán programarse las acciones correspondientes para que Zabbix sepa qué hacer con él. Desde el menú *Configuration -> Actions* podemos programar nuevas acciones cuyo origen sean reglas de descubrimiento.

Podremos decirle a Zabbix que, cada vez que detecte un nuevo equipo a través de reglas de descubrimiento, asigne a ese equipo un grupo o grupos y un listado de elementos de monitorización en forma de plantilla.

## 6.7. Inserción de parámetros de monitorización

Zabbix incluye por defecto una serie de plantillas (templates) con *items* y alertas predefinidas y que cumplen gran parte de las necesidades descritas en el análisis del sistema de videovigilancia. Sin embargo, hay necesidades específicas de este sistema y que supondrán la adición de nuevos parámetros de monitorización.

Las plantillas o *templates* por defecto incluidas en Zabbix que utilizaremos son dos:

- **Template Windows**: aplicaremos esta plantilla a todos aquellos equipos que tengan instalado el sistema operativo Windows.
- **Template Linux**: en este caso, los items definidos en esta plantilla son para la monitorización de equipos con sistema operativo Linux.

La idea es aprovechar las plantillas ya existentes y añadirles nuevos *items* e ir creando nuevas plantillas con los parámetros adicionales que necesitemos para la monitorización de los equipos del sistema de videovigilancia.

Crear una nueva plantilla es algo sencillo y que haremos desde el menú *Configuration -> Hosts*, seleccionando la opción "*Templates*" en el menú desplegable que aparece en la parte derecha del frontend de Zabbix.

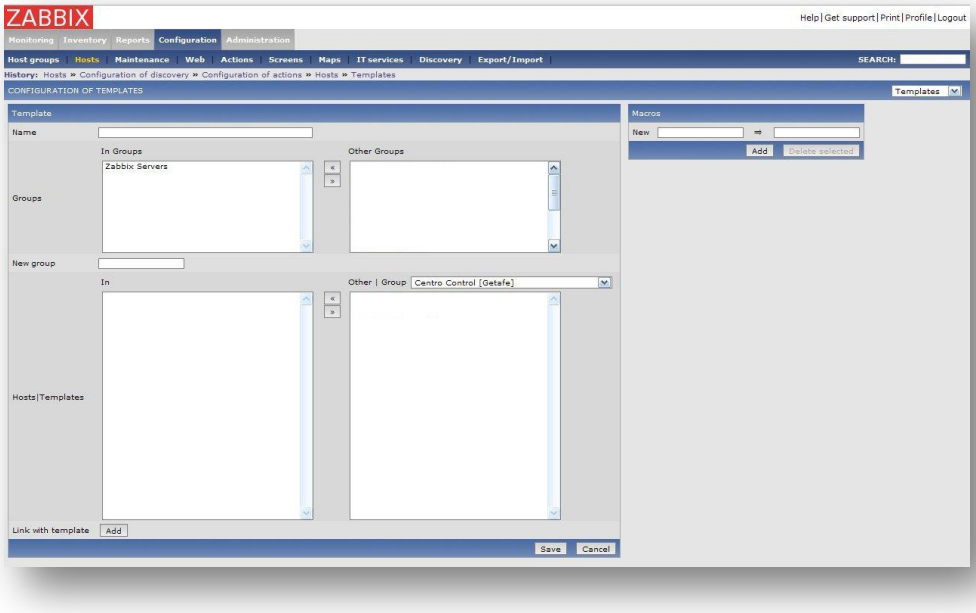



**Figura 29. Creación de plantillas de monitorización en Zabbix**

La inserción de nuevos parámetros de monitorización o *items* es también una tarea sencilla. Pongamos como ejemplo los servidores de grabación. Como su sistema operativo es Windows, aprovecharemos la plantilla "*Template Windows*". Supongamos que queremos monitorizar el porcentaje de espacio libre en la unidad C: en todos esos servidores (previamente, hemos de haber enlazado cada servidor con la plantilla mencionada). Accedemos a la plantilla "*Template Windows*" y creamos un nuevo *item*.

Se nos presenta entonces una pantalla en la que introduciremos los atributos del nuevo parámetro o *item* que queremos monitorizar. El atributo al que más atención debemos presentar es a la clave o *key*. La clave de los *items* es el dato que los diferencia entre ellos y lo que realmente indica **qué se está monitorizando**. Para más información sobre las claves soportadas por Zabbix, ver ANEXO II. Claves de monitorización en Zabbix.

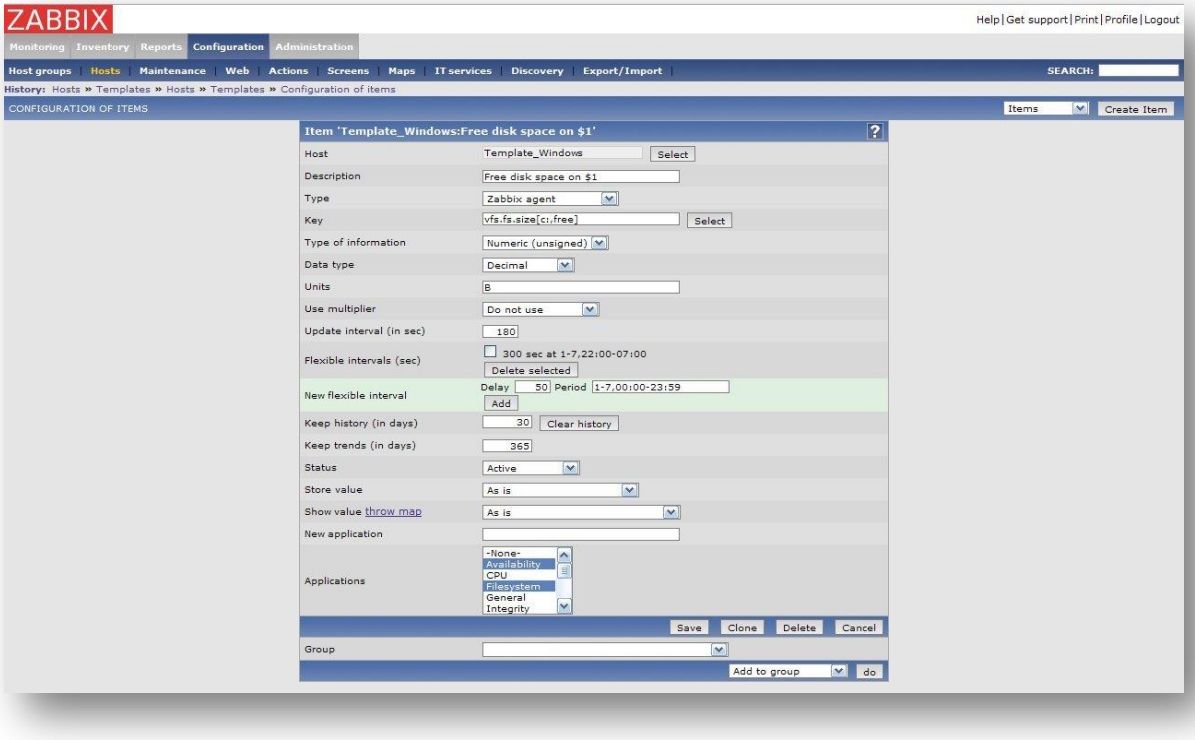



**Figura 30. Ejemplo de parámetro de monitorización en Zabbix**

- **Host:** aquí indicamos el nombre de la plantilla en la que incluiremos el ítem o del host al que lo aplicaremos.
- **Description (Descripción)**: se incluye aquí una descripción del parámetro de monitorización que se está creando.
- **Type (tipo)**: dado que el espacio en disco lo obtendremos a través del agente instalado en el host a monitorizar, el tipo escogido será "*Zabbix agent*".

- **Type of Information (tipo de información)**: será numérico sin signo.
- **Data type (tipo de dato)**: decimal.
- **Units (unidades)**: indicaremos que el dato que obtenemos viene dado en bytes (B).
- **Use multiplier (multiplicador)**: no utilizaremos multiplicador para el dato obtenido.
- **Update interval (intervalo de actualización)**: frecuencia de refresco (en segundo) del parámetro de monitorización.
- **Flexible intervals (intervalos de actualización flexibles)**: es posible indicar, para una determinada franja horaria, un intervalo de actualización distinto al establecido en el campo anterior.
- **Keep history (mantener historial)**: días durante los cuales mantendremos un histórico del parámetro.
- **Keep trends (datos de tendencias)**: días durante los cuales guardaremos los datos con los que se observa la evolución en la parámetro de monitorización.
- **Status (estado)**: lo crearemos como "activo".
- **Store value (almacenar valor)**: la opción "*As is*" indica que el valor se almacena tal cual se obtiene, sin hacer cálculos intermedios.
- **Show value (mostrar valor)**: es posible crear mapeados de valores (ver ANEXO IV. Mapeado de valores) para que así podamos crear correspondencias entre valores numéricos y textuales personalizados.
- **Applications (aplicaciones)**: aplicación o aplicaciones del parámetro creado.

### 6.7.1. Parámetros SNMP

Cuando se trata de monitorizar parámetros a través de SNMP existen ciertas diferencias con respecto a la monitorización vía agente Zabbix.

En el apartado de instalación del servidor central indicamos que Zabbix se compilaba con soporte SNMP, lo cual nos va a permitir monitorizar cualquier dispositivo que cuente con interfaz (agente) para este protocolo. Uno de los paquetes instalados como dependencias es ***snmp***, que nos provee de útiles herramientas con las cuales facilitar el trabajo con dispositivos que tengan habilitado el soporte a SNMP.

Como primer paso, verificaremos que el dispositivo que queremos monitorizar a través de SNMP responde a las peticiones que le enviemos. El paquete *snmp* nos permite comprobar la conectividad SNMP de un equipo escribiendo:

```
$ snmpstatus –v 2c –c public <IP address>
```

Si el demonio de snmp está correctamente arrancado y no hay problemas de conectividad, la salida del comando anterior debería ser la siguiente:

```
[UDP: [<IP address>]:161]=>[Hardware: x86 Family 6 Model 15 Stepping 13 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1 (Build 2600 Multiprocessor Free)] Up: 6:25:48.64
Interfaces: 2, Recv/Trans packets: 458500855/-1314337784 | IP: 457837705/-1315255457
```

Una vez comprobada la conectividad, es momento de obtener información del dispositivo a través de SNMP. Otro de los útiles comandos incluidos en el paquete *snmp* es ***snmpwalk*** [33]. Este comando intenta devolver todos los valores disponibles desde un agente SNMP concreto. Si escribimos

```
$ snmpwalk –v 2c –c public <IP address>
```

se mostrará un listado con todos esos valores. Dependiendo del dispositivo, esta lista puede ser muy extensa, así que, si lo deseamos, podemos restringir el número de valores que se nos mostrará por pantalla. Como ejemplo, vamos a mostrar las 6 primeras líneas de salida resultado de la ejecución del comando sobre un dispositivo SNMP concreto:

```
$ snmpwalk -v 2c -c public <IP address> | head -n 6

SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 6 Model 15 Stepping 13 AT/AT
COMPATIBLE - Software: Windows 2000 Version 5.1 (Build 2600 Multiprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.1
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (2328560) 6:28:05.60
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: XXXXX
SNMPv2-MIB::sysLocation.0 = STRING:
```

Como se puede ver en la salida, en la parte izquierda aparece el nombre o identificador y a la derecha, precedido del carácter **'='**, el valor obtenido. El identificador de la izquierda corresponde al OID (ver sección 3.3). Si ejecutamos de nuevo el comando con la opción "*-Ofn*", obtendremos ese identificador en formato numérico (con la rama completa de la MIB) en lugar de en formato textual.

```
$ snmpwalk -v 2c -c public –Ofn <IP address> | head -n 6

.1.3.6.1.2.1.1.1.0 = STRING: Hardware: x86 Family 6 Model 15 Stepping 13 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1 (Build 2600 Multiprocessor Free)
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.311.1.1.3.1.1
.1.3.6.1.2.1.1.3.0 = Timeticks: (2344045) 6:30:40.45
.1.3.6.1.2.1.1.4.0 = STRING:
.1.3.6.1.2.1.1.5.0 = STRING: XXXXX
.1.3.6.1.2.1.1.6.0 = STRING:
```

Vamos a utilizar uno de esos OID para crear un ejemplo de parámetro monitorizado con SNMP. Monitorizaremos el nombre del sistema, dado por el OID ***sysDescr.0***. A la hora de crear el *item* que monitoriza este parámetro, navegamos a la plantilla o equipo en el que deseemos crearlo (*Configuration -> Hosts -> Crear item*) y se nos mostrará la siguiente pantalla:

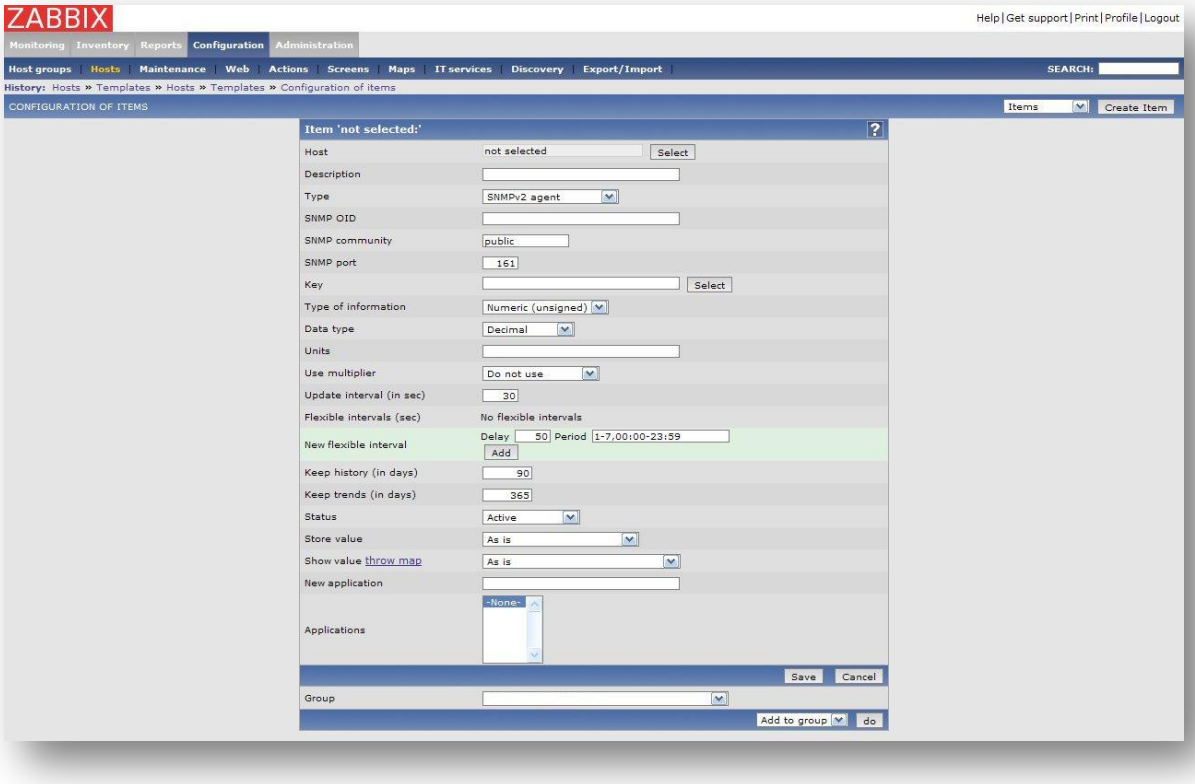



**Figura 31. Creación de parámetros SNMP en Zabbix**

Los valores que introduciremos para cada campo son:

- **Description (Descripción)**: escribiremos "System name".

- **Type (Tipo)**: seleccionaremos la versión del agente SNMP (en este caso SNMPv2).
- **SNMP community (comunidad SNMP)**: mantenemos el valor por defecto, "public".
- **SNMP-OID (identificador del objeto SNMP)**: aquí es donde escribiremos el identificador (en formato texto o numérico) que obtuvimos al ejecutar el comando *snmpwalk*. En este caso, el valor es "SNMPv2-MIB::sysDescr.0".
- **Key (clave)**: escribiremos "sysDescr.0" para este campo.
- **Type of information (tipo de información)**: seleccionamos "Character" (valor de tipo texto).
- **Update interval (intervalo de actualización)**: el nombre del sistema no es algo que vaya a cambiar cada poco tiempo, así que introduciremos un intervalo alto, como por ejemplo "86400".

Existen ejemplos concretos de parámetros de monitorización cuyos valores se deben obtener convenientemente a través de SNMP. Tal es el caso del tráfico de red para dispositivos como conmutadores o enrutadores. Sin embargo, nos encontramos con un problema. A la hora de monitorizar el tráfico de red, tenemos disponibles varias interfaces y el descriptor de cada una de éstas varía según el tipo de dispositivo.

Ejecutemos el comando *snmpwalk* sobre uno de los equipos en los que queramos monitorizar el tráfico de red. En la salida del comando encontraremos información sobre las interfaces de red del equipo:

```
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
```

Vemos que el equipo cuenta con dos interfaces, "*lo*" y "*eth0*" y que los índices para cada una de ellas son, respectivamente, "1" y "2". La salida del comando nos ofrece además los datos de tráfico de red para esas dos interfaces:

```
IF-MIB::ifOutOctets.1 = Counter32: 1825596052
IF-MIB::ifOutOctets.2 = Counter32: 1533857263
```

Si queremos monitorizar el tráfico de salida en la interfaz *eth0*, teóricamente nos valdría con crear un item SNMP que tuviera por clave "*IF-MIB::ifOutOctets.2*". Sin embargo, no deberíamos, ya que mientras que en este equipo que hemos considerado el tráfico de salida para *eth0* sí que corresponde a esa clave, para otro equipo el índice puede variar y la clave ya no nos valdría. Por ejemplo, en lugar de "2", el índice para *eth0* podría ser "3", con lo cual la clave a escribir sería "*IF-MIB::ifOutOctets.3*".

Es por ello que tendremos consideraciones especiales a la hora de monitorizar parámetros de monitorización de este tipo en los que los índices de los OID puden ser dinámicos.

Así, si queremos crear un *item* en Zabbix que monitorice ese tráfico de salida, escribiríamos los siguientes valores en el formulario de creación:

- **Description (Descripción):** "Tráfico de salida en la interfaz $1". El valor "$1" es una referencia al parámetro que pasaremos a Zabbix a través de la clave (será *eth0* en este caso).
- **Type (tipo):** SNMPv2 agent (versión 2 del agente SNMP).
- **SNMP OID:** escribiremos "*IF-MIB::ifOutOctets["index","ifDescr","eth0"]*".
- **Key (clave):** será "ifOutOctets[eth0]". El valor encerrado entre corchetes es el valor que tomará $1 visto en la descripción del item.
- **Type of information (tipo de valor):** seleccionaremos "Numeric (unsigned)", es decir, numérico sin signo.
- **Units (unidades):** Bps (bytes por segundo).
- **Store value (guardar valor):** lo almacenaremos como una delta que indica el cambio por segundo en el valor del item.

Crearemos un item análogo al anterior para cada una de las interfaces que tenga el equipo monitorizado y con ello solucionaremos el problema de los índices dinámicos.

### 6.7.2. Parámetros de monitorización en los servidores de grabación

Mostraremos aquí un conjunto de tablas en las que aparezcan las necesidades para los servidores de grabación (expuestas en el punto 4.1.5) implementadas en forma de los correspondientes *items* de Zabbix y con la plantilla en la que éstos se incluyen.

## ❶ Parámetros agente Zabbix

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | | |
| Información del host | Nombre del equipo, sistema operativo, fabricante del procesador | Host information | Zabbix agent | system.uname | (no aplica) | As is | General | Template Windows |
| | Estado del equipo | Host status | Zabbix agent | icmpping[,,2000,,] | (no aplica) | As is | Availability | Template Windows |
| | Tiempo en funcionamiento | Host uptime | Zabbix agent | system.uptime | uptime | As is | General | Template Windows |
| **DISPONIBILIDAD** | | | | | | | | |
| Almacenamiento | Espacio libre en la unidad C: (bytes) | Free disk space on $1 | Zabbix agent | vfs.fs.size[c:,free] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| | Espacio libre en la unidad C: (%) | Free disk space on $1 (in %) | Zabbix agent | vfs.fs.size[c:,pfree] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| | Espacio total en la unidad C: (bytes) | Total disk space on $1 | Zabbix agent | vfs.fs.size[c:,total] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| **MEMORIA FÍSICA Y MEMORIA SWAP** | | | | | | | | |
| Memoria física | Memoria física libre | Free memory | Zabbix agent | vm.memory.size[free] | B (bytes) | As is | Availability, Memory | Template Windows |
| | Memoria física total | Total memory | Zabbix agent | vm.memory.size[total] | B (bytes) | As is | Availability, Memory | Template Windows |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Memoria swap | Memoria swap libre | Free swap space | Zabbix agent | system.swap.size[,free] | B (bytes) | As is | Availability, Memory | Template Windows |
| | Memoria swap total | Total swap space | Zabbix agent | system.swap.size[,total] | B (bytes) | As is | Availability, Memory | Template Windows |
| **RENDIMIENTO DE LA CPU** | | | | | | | | |
| Carga del procesador | | Processor load | Zabbix agent | system.cpu.load[,avg1] | (no aplica) | As is | CPU, Performance | Template Windows |
| Media de carga del procesador en los últimos 5 min. | | Processor load5 | Zabbix agent | system.cpu.load[,avg5] | (no aplica) | As is | CPU, Performance | Template Windows |
| Media de carga del procesador en los últimos 15 min. | | Processor load15 | Zabbix agent | system.cpu.load[,avg15] | (no aplica) | As is | CPU, Performance | Template Windows |
| Porcentaje de utilización del procesador | | CPU usage (in %) | Zabbix agent | system.cpu.util[„] | (no aplica) | As is | CPU, Performance | Template Windows |
| **INTEGRIDAD DE FICHEROS** | | | | | | | | |
| Comprobación del checksum del fichero autoexec.bat | | Checksum of autoexec.bat | Zabbix agent | vfs.file.cksum[c:\autoexec.bat] | (no aplica) | As is | Integrity | Template Windows |
| Comprobación del checksum del fichero config.sys | | Checksum of config.sys | Zabbix agent | vfs.file.cksum[c:\config.sys] | (no aplica) | As is | Integrity | Template Windows |
| **PROCESOS** | | | | | | | | |
| Número de procesos ejecutándose en el sistema | | Number of processes | Zabbix agent | proc.num[] | (no aplica) | As is | Processes | Template Windows |
| **SERVICIOS Y APLICACIONES** | | | | | | | | |
| Número de procesos Sony RealShot Manager ejecutándose | | Number of running Sony RSM processes | Zabbix agent | proc.num["RealShot Manager.exe"] | (no aplica) | As is | Processes | Template Windows |
| Número de procesos Zabbix Agent ejecutándose | | Number of running zabbix agentd processes | Zabbix agent | proc.num[“zabbix_agentd.exe”] | (no aplica) | As is | Processes | Template Windows |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Estado del servicio SNMP de Windows | SNMP agent service state | Zabbix agent | service_state[SNMP] | (no aplica) | As is | Services | Template Windows |
| Estado del servicio DHCP | DHCP client service state (Dhcp) | Zabbix agent | service_state[Dhcp] | (no aplica) | As is | Services | Template Windows |
| Estado del servicio del agente Zabbix | Zabbix agent service state | Zabbix agent | service_state[Zabbix Agent] | (no aplica) | As is | Services | Template Windows |
| Versión del agente Zabbix instalado | Version of zabbix_agent (d) running | Zabbix agent | agent.version | (no aplica) | As is | General | Template Windows |
| **RED** | | | | | | | |
| Conexiones TCP activas | Active TCP connections | Zabbix agent | perf_counter[\638\644] | (no aplica) | As is | Network | Template Windows |
| Conexiones TCP establecidas | Established TCP connections | Zabbix agent | perf_counter[\638\642] | (no aplica) | As is | Network | Template Windows |
| Ping TCP al agente Zabbix | Ping to Zabbix Agent | Zabbix agent | agent.ping | (no aplica) | As is | Network | Template Windows |
| Bytes recibidos por segundo (tráfico de red de entrada) | Incoming traffic | Zabbix agent | NetIn | Bps | Delta (speed per second) | Network | Template Windows |
| Bytes transmitidos por segundo (tráfico de red de salida) | Outgoing traffic | Zabbix agent | NetOut | Bps | Delta (speed per second) | Network | Template Windows |
| **WINDOWS LOGS** | | | | | | | |
| Log de Aplicación | Application Log | Zabbix agent (active) | eventlog[Application] | (no aplica) | As is | Windows Logs | Windows Logging |
| Log de Sistema | System Log | Zabbix agent (active) | eventlog[System] | (no aplica) | As is | Windows Logs | Windows Logging |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Log de Seguridad | Security Log | Zabbix agent (active) | eventlog[Security] | (no aplica) | As is | Windows Logs | Windows Logging |

### ❷ Parámetros SNMP

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **DISPONIBILIDAD** | | | | | | | | |
| Almacenamiento | Capacidad del sistema RAID (en Mbytes) | RAID Unit capacity (MB) | SNMPv2 agent | enterprises.1458.100.23.1.6.1 | (no aplica) | As is | RAID check | Template RAID |
| | Estado de la unidad RAID | RAID Unit status | SNMPv2 agent | enterprises.1458.100.23.1.7.1 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #0 | Drive #0 model | SNMPv2 agent | enterprises.1458.100.22.1.6.1 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #1 | Drive #1 model | SNMPv2 agent | enterprises.1458.100.22.1.6.2 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #2 | Drive #2 model | SNMPv2 agent | enterprises.1458.100.22.1.6.3 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #3 | Drive #3 model | SNMPv2 agent | enterprises.1458.100.22.1.6.4 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #4 | Drive #4 model | SNMPv2 agent | enterprises.1458.100.22.1.6.5 | (no aplica) | As is | RAID check | Template RAID |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Almacenamiento | Modelo del disco duro #5 | Drive #5 model | SNMPv2 agent | enterprises.1458.100.22.1.6.6 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #6 | Drive #6 model | SNMPv2 agent | enterprises.1458.100.22.1.6.7 | (no aplica) | As is | RAID check | Template RAID |
| | Modelo del disco duro #7 | Drive #7 model | SNMPv2 agent | enterprises.1458.100.22.1.6.8 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #0 | Drive #0 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.1 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #1 | Drive #1 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.2 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #2 | Drive #2 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.3 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #3 | Drive #3 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.4 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #4 | Drive #4 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.5 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #5 | Drive #5 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.6 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #6 | Drive #6 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.7 | (no aplica) | As is | RAID check | Template RAID |
| | Temperatura del disco duro #7 | Drive #7 Temperature | SNMPv2 agent | enterprises.1458.100.22.1.15.8 | (no aplica) | As is | RAID check | Template RAID |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Almacenamiento | Tiempo en funcionamiento del disco #0 | Drive #0 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.1 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #1 | Drive #1 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.2 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #2 | Drive #2 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.3 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #3 | Drive #3 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.4 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #4 | Drive #4 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.5 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #5 | Drive #5 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.6 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #6 | Drive #6 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.7 | (no aplica) | As is | RAID check | Template RAID |
| | Tiempo en funcionamiento del disco #7 | Drive #7 power (in days) | SNMPv2 agent | enterprises.1458.100.22.1.16.8 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #0 | Drive #0 status | SNMPv2 agent | enterprises.1458.100.22.1.10.1 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #1 | Drive #1 status | SNMPv2 agent | enterprises.1458.100.22.1.10.2 | (no aplica) | As is | RAID check | Template RAID |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Almacenamiento | Estado del disco #2 | Drive #2 status | SNMPv2 agent | enterprises.1458.100.22.1.10.3 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #3 | Drive #3 status | SNMPv2 agent | enterprises.1458.100.22.1.10.4 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #4 | Drive #4 status | SNMPv2 agent | enterprises.1458.100.22.1.10.5 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #5 | Drive #5 status | SNMPv2 agent | enterprises.1458.100.22.1.10.6 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #6 | Drive #6 status | SNMPv2 agent | enterprises.1458.100.22.1.10.7 | (no aplica) | As is | RAID check | Template RAID |
| | Estado del disco #7 | Drive #7 status | SNMPv2 agent | enterprises.1458.100.22.1.10.8 | (no aplica) | As is | RAID check | Template RAID |
| | Número de discos conectados a la controladora RAID | Number of drives attached to the controller | SNMPv2 agent | enterprises.1458.100.20.1.11.1 | (no aplica) | As is | RAID check | Template RAID |
| | Porcentaje completado en la unidad RAID | RAID Unit percentage completed | SNMPv2 agent | enterprises.1458.100.23.1.10.1 | (no aplica) | As is | RAID check | Template RAID |
| RED | | | | | | | | |
| Bytes recibidos por segundo (tráfico de entrada) | | Received bytes | SNMPv2 agent | enterprises.343.2.7.2.2.1.3.1.3.8 | Bps (bytes por segundo) | Delta (speed per second) | Network | Template Windows |
| Bytes transmitidos por segundo (tráfico de salida) | | Transmitted bytes | SNMPv2 agent | enterprises.343.2.7.2.2.1.3.1.4.8 | Bps (bytes por segundo) | Delta (speed per second) | Network | Template Windows |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **REFRIGERACIÓN** | | | | | | | |
| Temperatura de la CPU | CPU Temperature | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.15 | °C | As is | Temperature | Template Supermicro |
| Temperatura del sistema | System Temperature | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.16 | °C | As is | Temperature | Template Supermicro |
| Velocidad del ventilador #1 | Fan1 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.1 | (no aplica) | As is | Cooling | Template Supermicro |
| Velocidad del ventilador #2 | Fan2 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.2 | (no aplica) | As is | Cooling | Template Supermicro |
| Velocidad del ventilador #3 | Fan3 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.3 | (no aplica) | As is | Cooling | Template Supermicro |
| Velocidad del ventilador #4 | Fan4 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.4 | (no aplica) | As is | Cooling | Template Supermicro |
| Velocidad del ventilador #5 | Fan5 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.5 | (no aplica) | As is | Cooling | Template Supermicro |
| Velocidad del ventilador #6 | Fan6 Fan Speed (in rpm) | SNMPv2 agent | enterprises.10876.2.1.1.1.1.4.6 | (no aplica) | As is | Cooling | Template Supermicro |

### ❸ Comprobaciones sencillas (simple checks)

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **RED** | | | | | | | |
| Ping TCP al servidor | Ping to the server (TCP) | Simple check | icmpping[„500„] | (no aplica) | As is | Network | Template Windows |
| Estado del servidor HTTP | HTTP server status | Simple check | http,80 | (no aplica) | As is | Network | Template Windows |

## 6.7.1. Parámetros de monitorización en las cámaras de videovigilancia

Las cámaras se monitorizan a través de parámetros obtenidos vía SNMP y comprobaciones sencillas (simple checks).

### ❶ Parámetros SNMP

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | |
| Modelo de cámara | Camera model | SNMPv2 agent | sysDescr.0 | (no aplica) | As is | General | Template Cameras |
| Versión del firmware instalado | Firmware version | SNMPv2 agent | enterprises.122.8501.2.1.1.1.8.1 | (no aplica) | As is | General | Template Cameras |
| Número de serie | Serial number | SNMPv2 agent | enterprises.122.8501.2.1.1.1.6.1 | (no aplica) | As is | General | Template Cameras |
| Tráfico de red de entrada | Incoming traffic | SNMPv2 agent | ifInOctets.2 | Bps (bytes por segundo) | Delta (speed per second) | Network | Template Cameras |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Tráfico de red de salida | Outgoing traffic | SNMPv2 agent | ifOutOctets.2 | Bps (bytes por segundo) | Delta (speed per second) | Network | Template Cameras |

### ❷ Comprobaciones sencillas (simple checks)

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| INFORMACIÓN GENERAL | | | | | | | |
| Estado del servidor HTTP | HTTP server status | Simple check | http,80 | (no aplica) | As is | Network | Template Cameras |
| Respuesta a ping | Camera response | Simple check | Icmpping[,3,500,,] | (no aplica) | As is | Network | Template Cameras |

## 6.7.2. Parámetros de monitorización en equipos de los centros de control

Los equipos de los centros de control se monitorizan fundamentalmente a través de un agente Zabbix instalado en ellos.

### ❶ Parámetros agente Zabbix

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | | |
| Información del host | Nombre del equipo, sistema operativo, fabricante del procesador | Host information | Zabbix agent | system.uname | (no aplica) | As is | General | Template Windows |
| | Estado del equipo | Host status | Zabbix agent | icmpping[,,2000,,] | (no aplica) | As is | Availability | Template Windows |
| | Tiempo en funcionamiento | Host uptime | Zabbix agent | system.uptime | uptime | As is | General | Template Windows |
| **DISPONIBILIDAD** | | | | | | | | |
| Almacenamiento | Espacio libre en la unidad C: (bytes) | Free disk space on $1 | Zabbix agent | vfs.fs.size[c:,free] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| | Espacio libre en la unidad C: (%) | Free disk space on $1 (in %) | Zabbix agent | vfs.fs.size[c:,pfree] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| | Espacio total en la unidad C: (bytes) | Total disk space on $1 | Zabbix agent | vfs.fs.size[c:,total] | B (bytes) | As is | Availability, Filesystem | Template Windows |
| **MEMORIA FÍSICA Y MEMORIA SWAP** | | | | | | | | |
| Memoria física | Memoria física libre | Free memory | Zabbix agent | vm.memory.size[free] | B (bytes) | As is | Availability, Memory | Template Windows |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Memoria física | Memoria física total | Total memory | Zabbix agent | vm.memory.size[total] | B (bytes) | As is | Availability, Memory | Template Windows |
| Memoria swap | Memoria swap libre | Free swap space | Zabbix agent | system.swap.size[,free] | B (bytes) | As is | Availability, Memory | Template Windows |
| | Memoria swap total | Total swap space | Zabbix agent | system.swap.size[,total] | B (bytes) | As is | Availability, Memory | Template Windows |
| RENDIMIENTO DE LA CPU | | | | | | | | |
| Carga del procesador | | Processor load | Zabbix agent | system.cpu.load[,avg1] | (no aplica) | As is | CPU, Performance | Template Windows |
| Media de carga del procesador en los últimos 5 min. | | Processor load5 | Zabbix agent | system.cpu.load[,avg5] | (no aplica) | As is | CPU, Performance | Template Windows |
| Media de carga del procesador en los últimos 15 min. | | Processor load15 | Zabbix agent | system.cpu.load[,avg15] | (no aplica) | As is | CPU, Performance | Template Windows |
| Porcentaje de utilización del procesador | | CPU usage (in %) | Zabbix agent | system.cpu.util[„] | (no aplica) | As is | CPU, Performance | Template Windows |
| INTEGRIDAD DE FICHEROS | | | | | | | | |
| Comprobación del checksum del fichero autoexec.bat | | Checksum of autoexec.bat | Zabbix agent | vfs.file.cksum[c:\autoexec.bat] | (no aplica) | As is | Integrity | Template Windows |
| Comprobación del checksum del fichero config.sys | | Checksum of config.sys | Zabbix agent | vfs.file.cksum[c:\config.sys] | (no aplica) | As is | Integrity | Template Windows |
| PROCESOS | | | | | | | | |
| Número de procesos ejecutándose en el sistema | | Number of processes | Zabbix agent | proc.num[] | (no aplica) | As is | Processes | Template Windows |
| SERVICIOS Y APLICACIONES | | | | | | | | |
| Número de procesos Sony RealShot Manager ejecutándose | | Number of running Sony RSM processes | Zabbix agent | proc.num["RealShot Manager.exe"] | (no aplica) | As is | Processes | Template Windows |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Número de procesos Zabbix Agent ejecutándose | Number of running zabbix agentd processes | Zabbix agent | proc.num[“zabbix_agentd.exe”] | (no aplica) | As is | Processes | Template Windows |
| Estado del servicio SNMP de Windows | SNMP agent service state | Zabbix agent | service_state[SNMP] | (no aplica) | As is | Services | Template Windows |
| Estado del servicio DHCP | DHCP client service state (Dhcp) | Zabbix agent | service_state[Dhcp] | (no aplica) | As is | Services | Template Windows |
| Estado del servicio del agente Zabbix | Zabbix agent service state | Zabbix agent | service_state[Zabbix Agent] | (no aplica) | As is | Services | Template Windows |
| Versión del agente Zabbix instalado | Version of zabbix_agent (d) running | Zabbix agent | agent.version | (no aplica) | As is | General | Template Windows |
| **RED** | | | | | | | |
| Conexiones TCP activas | Active TCP connections | Zabbix agent | perf_counter[\638\644] | (no aplica) | As is | Network | Template Windows |
| Conexiones TCP establecidas | Established TCP connections | Zabbix agent | perf_counter[\638\642] | (no aplica) | As is | Network | Template Windows |
| Ping TCP al agente Zabbix | Ping to Zabbix Agent | Zabbix agent | agent.ping | (no aplica) | As is | Network | Template Windows |
| Bytes recibidos por segundo (tráfico de red de entrada) | Incoming traffic | Zabbix agent | NetIn | Bps | Delta (speed per second) | Network | Template Windows |
| Bytes transmitidos por segundo (tráfico de red de salida) | Outgoing traffic | Zabbix agent | NetOut | Bps | Delta (speed per second) | Network | Template Windows |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **WINDOWS LOGS** | | | | | | | |
| Log de Aplicación | Application Log | Zabbix agent (active) | eventlog[Application] | (no aplica) | As is | Windows Logs | Windows Logging |
| Log de Sistema | System Log | Zabbix agent (active) | eventlog[System] | (no aplica) | As is | Windows Logs | Windows Logging |
| Log de Seguridad | Security Log | Zabbix agent (active) | eventlog[Security] | (no aplica) | As is | Windows Logs | Windows Logging |

### 6.7.3. Parámetros de monitorización en los conmutadores centrales de la red CCTV

La monitorización de toda la electrónica de red se lleva a cabo a través de parámetros SNMP.

#### ❶ Parámetros SNMP

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | |
| Ubicación del conmutador | Device location | SNMPv2 agent | sysLocation.0 | (no aplica) | As is | General | Template Cisco Catalyst 4507R |
| Modelo del conmutador | Device model | SNMPv2 agent | mib-2.47.1.1.1.1.2.1 | (no aplica) | As is | General | Template Cisco Catalyst 4507R |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Nombre del conmutador | | Device name | SNMPv2 agent | sysName.0 | (no aplica) | As is | General | Template Cisco Catalyst 4507R |
| Tiempo que lleva en funcionamiento el conmutador | | Uptime | SNMPv2 agent | sysUpTimeInstance | (no aplica) | As is | General | Template Cisco Catalyst 4507R |
| RED | | | | | | | | |
| Información de cada puerto | Estado administrativo de la interfaz | ifAdminStatus | SNMPv2 agent | ifAdminStatus.X (*1) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |
| | Estado operativo de la interfaz | ifOperStatus | SNMPv2 agent | ifOperStatus.X (*1[17]) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |
| | Nombre alias para la interfaz | ifAlias | SNMPv2 agent | ifAlias.X (*1) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |
| | Descriptor de la interfaz | ifDescr | SNMPv2 agent | ifDescr.X (*1) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |
| | Bytes de entrada a la interfaz | ifInOctets | SNMPv2 agent | ifInOctets.X (*1) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |
| | Bytes de salida desde la interfaz | ifOutOctets | SNMPv2 agent | ifOutOctets.X (*1) | (no aplica) | As is | Network | Template Cisco Catalyst 4507R |

[17] ***1**, X es cualquier puerto del conmutador

## 6.7.4. Parámetros de monitorización en los conmutadores centrales de campus en la red UC3M

### ❶ Parámetros SNMP

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | | |
| Tiempo que lleva en funcionamiento el conmutador | | Uptime | SNMPv2 agent | sysUpTimeInstance | (no aplica) | As is | General | Template Cisco6509-50G |
| Carga de CPU | | CPU usage | SNMPv2 agent | cpuUsage | (no aplica) | As is | CPU | Template Cisco6509-50G |
| **RED** | | | | | | | | |
| Información de cada puerto | Bytes de entrada a la interfaz | ifInOctets | SNMPv2 agent | ifInOctets.X (*2[18]) | (no aplica) | As is | Network | Template Cisco6509-50G |
| | Bytes de salida desde la interfaz | ifOutOctets | SNMPv2 agent | ifOutOctets.X (*2) | (no aplica) | As is | Network | Template Cisco6509-50G |

[18] ***2**, X es cualquier puerto del conmutador

## 6.7.5. Parámetros de monitorización en los conmutadores de planta de los campus

### ❶ Parámetros SNMP

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | |
| Ubicación del conmutador | Device location | SNMPv2 agent | sysLocation.0 | (no aplica) | As is | General | Template Cisco Catalyst_2960 24TC-L |
| Modelo del conmutador | Device model | SNMPv2 agent | mib-2.47.1.1.1.1.2.1 | (no aplica) | As is | General | Template Cisco Catalyst_2960 24TC-L |
| Nombre del conmutador | Device name | SNMPv2 agent | sysName.0 | (no aplica) | As is | General | Template Cisco Catalyst_2960 24TC-L |
| Tiempo que lleva en funcionamiento el conmutador | Uptime | SNMPv2 agent | sysUpTimeInstance | (no aplica) | As is | General | Template Cisco Catalyst_2960 24TC-L |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **RED** | | | | | | | | |
| Información de cada puerto | Estado administrativo de la interfaz | ifAdminStatus | SNMPv2 agent | ifAdminStatus.X (*3[19]) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |
| | Estado operativo de la interfaz | ifOperStatus | SNMPv2 agent | ifOperStatus.X (*3) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |
| | Nombre alias para la interfaz | ifAlias | SNMPv2 agent | ifAlias.X (*3) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |
| | Descriptor de la interfaz | ifDescr | SNMPv2 agent | ifDescr.X (*3) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |
| | Bytes de entrada a la interfaz | ifInOctets | SNMPv2 agent | ifInOctets.X (*3) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |
| | Bytes de salida desde la interfaz | ifOutOctets | SNMPv2 agent | ifOutOctets.X (*3) | (no aplica) | As is | Network | Template Cisco Catalyst_2960 24TC-L |

[19] ***3**, X es cualquier puerto del conmutador

## 6.7.6. Parámetros de monitorización en el servidor Zabbix

Se decide monitorizar también el servidor central Zabbix a través del agente instalado al efecto.

### ❶ Parámetros agente Zabbix

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **INFORMACIÓN GENERAL** | | | | | | | | |
| Información del host | Nombre del equipo, sistema operativo, fabricante del procesador | Host information | Zabbix agent | system.uname | (no aplica) | As is | General | Template Linux |
| | Estado del equipo | Host status | Zabbix agent | status | (no aplica) | As is | Availability | Template Linux |
| | Tiempo en funcionamiento | Host uptime | Zabbix agent | system.uptime | uptime | As is | General | Template Linux |
| | Nombre del equipo | Host name | Zabbix agent | system.hostname | (no aplica) | As is | General | Template Linux |
| | Número de usuarios conectados | Number of Users connected | Zabbix agent | system.users.num | (no aplica) | As is | General | Template Linux |
| **DISPONIBILIDAD** | | | | | | | | |
| Almacenamiento | Espacio libre en / (bytes) | Free disk space on / | Zabbix agent | vfs.fs.size[/,free] | B (bytes) | As is | Availability, Filesystem | Template Linux |
| | Espacio libre en / (%) | Free disk space on / (in %) | Zabbix agent | vfs.fs.size[/,pfree] | B (bytes) | As is | Availability, Filesystem | Template Linux |

| Parámetro | | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| **MEMORIA FÍSICA Y MEMORIA SWAP** | | | | | | | | |
| Memoria física | Memoria física libre | Free memory | Zabbix agent | vm.memory.size[free] | B (bytes) | As is | Availability, Memory | Template Linux |
| | Memoria física total | Total memory | Zabbix agent | vm.memory.size[total] | B (bytes) | As is | Availability, Memory | Template Linux |
| Memoria swap | Memoria swap libre | Free swap space | Zabbix agent | system.swap.size[,free] | B (bytes) | As is | Availability, Memory | Template Linux |
| | Memoria swap total | Total swap space | Zabbix agent | system.swap.size[,total] | B (bytes) | As is | Availability, Memory | Template Linux |
| **RENDIMIENTO DE LA CPU** | | | | | | | | |
| Carga de CPU | | CPU load | Zabbix agent | system.cpu.load | (no aplica) | As is | CPU, Performance | Template Linux |
| Carga del procesador | | Processor load | Zabbix agent | system.cpu.load[,avg1] | (no aplica) | As is | CPU, Performance | Template Linux |
| Media de carga del procesador en los últimos 5 min. | | Processor load5 | Zabbix agent | system.cpu.load[,avg5] | (no aplica) | As is | CPU, Performance | Template Linux |
| Media de carga del procesador en los últimos 15 min. | | Processor load15 | Zabbix agent | system.cpu.load[,avg15] | (no aplica) | As is | CPU, Performance | Template Linux |
| Porcentaje de utilización del procesador | | CPU usage (in %) | Zabbix agent | system.cpu.util[,,] | (no aplica) | As is | CPU, Performance | Template Linux |
| **PROCESOS** | | | | | | | | |
| Número de procesos ejecutándose en el sistema | | Number of processes | Zabbix agent | proc.num[] | (no aplica) | As is | Processes | Template Linux |
| **SERVICIOS Y APLICACIONES** | | | | | | | | |
| Número de procesos zabbix_agentd ejecutándose | | Number of running processes zabbix_agentd | Zabbix agent | proc.num[zabbix_agentd] | (no aplica) | As is | Processes | Template Linux |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Número de procesos zabbix_server ejecutándose | Number of running processes zabbix_server | Zabbix agent | proc.num[zabbix_server] | (no aplica) | As is | Processes | Template Linux |
| Número de procesos apache ejecutándose | Number of running processes apache | Zabbix agent | proc.num[apache2] | (no aplica) | As is | Processes | Template Linux |
| Número de procesos mysql ejecutándose | Number of running processes mysqld | Zabbix agent | proc.num[mysqld] | (no aplica) | As is | Processes | Template Linux |
| Número de procesos inetd ejecutándose | Number of running processes inetd | Zabbix agent | proc.num[inetd] | (no aplica) | As is | Processes | Template Linux |
| Número de procesos sshd ejecutándose | Number of running processes sshd | Zabbix agent | proc.num[sshd] | (no aplica) | As is | Processes | Template Linux |
| Número de procesos syslogd ejecutándose | Number of running processes syslogd | Zabbix agent | proc.num[syslogd] | (no aplica) | As is | Processes | Template Linux |
| Estado del servicio del agente Zabbix | Zabbix agent service state | Zabbix agent | service_state[Zabbix Agent] | (no aplica) | As is | Services | Template Linux |
| Estado del servidor SSH | SSH server is running | Zabbix agent | net.tcp.service[ssh] | (no aplica) | As is | Services | Template Linux |
| Estado del servidor HTTP | WEB (http) server is running | Zabbix agent | net.tcp.service[http] | (no aplica) | As is | Services | Template Linux |
| **RED** | | | | | | | |
| Ping TCP al agente Zabbix | Ping to Zabbix Agent | Zabbix agent | agent.ping | (no aplica) | As is | Network | Template Linux |

| Parámetro | Zabbix item | | | | | | |
|---|---|---|---|---|---|---|---|
| | Descripción | Tipo | Clave | Unidad | Almacenar Valor | Aplicación | Plantilla |
| Tráfico de entrada en la interfaz eth0 | Incoming traffic on interface eth0 | Zabbix agent | net.if.in[eth0,bytes] | Bps | Delta (speed per second) | Network | Template Linux |
| Tráfico de entrada en la interfaz eth1 | Incoming traffic on interface eth1 | Zabbix agent | net.if.in[eth1,bytes] | Bps | Delta (speed per second) | Network | Template Linux |
| Tráfico de salida en la interfaz eth0 | Incoming traffic on interface eth0 | Zabbix agent | net.if.out[eth0,bytes] | Bps | Delta (speed per second) | Network | Template Linux |
| Tráfico de salida en la interfaz eth1 | Incoming traffic on interface eth1 | Zabbix agent | net.if.out[eth1,bytes] | Bps | Delta (speed per second) | Network | Template Linux |
| INTEGRIDAD DE FICHEROS | | | | | | | |
| Comprobación del checksum del fichero /etc/inetd.conf | Checksum of /etc/inetd.conf | Zabbix agent | vfs.file.cksum[/etc/inetd.conf] | (no aplica) | As is | Integrity | Template Linux |
| Comprobación del checksum del fichero /etc/services | Checksum of /etc/services | Zabbix agent | vfs.file.cksum[/etc/services] | (no aplica) | As is | Integrity | Template Linux |
| Comprobación del checksum del fichero /usr/bin/sshd | Checksum of /usr/bin/sshd | Zabbix agent | vfs.file.cksum[/usr/bin/sshd] | (no aplica) | As is | Integrity | Template Linux |
| Comprobación del checksum del fichero /usr/bin/ssh | Checksum of /usr/bin/ssh | Zabbix agent | vfs.file.cksum[/usr/bin/ssh] | (no aplica) | As is | Integrity | Template Linux |
| Comprobación del checksum del fichero /etc/passwd | Checksum of /etc/passwd | Zabbix agent | vfs.file.cksum[/etc/passwd] | (no aplica) | As is | Integrity | Template Linux |

## 6.8. Creación de alertas

Una alerta en Zabbix se produce como resultado de un disparador o iniciador que se activa cuando se registran ciertas condiciones en los valores de monitorización.

En la sección 5.5 enumeramos los distintos niveles de alerta y las alertas correspondientes a cada uno de esos niveles. Explicaremos a continuación cómo se han implementado esas alertas en términos de los *ítems* de Zabbix implicados.

### 6.8.1. Nivel de "Información" (Information)

- **Aviso de reinicio en el servidor de Zabbix**: cada vez que un equipo se reinicia, el valor que toma el tiempo que lleva en funcionamiento pasa a valer cero. El ítem de Zabbix implicado sería "Host uptime" y si su valor es menor a 5 minutos, consideraremos que acaba de ser reiniciado. En caso de que así sea, se enviará un correo electrónico informando del evento.
- **Cambios en la configuración del servidor Zabbix:** se evalúa si el último valor recibido para el ítem "Host Information" es diferente.
- **Cambio en el nombre del servidor Zabbix**: esta alerta se programa para detectar modificaciones en el valor del ítem "Host name".
- **Comienzo y finalización de las operaciones de verificación y reconstrucción de los sistemas RAID de almacenamiento en los servidores de grabación**: los ítems a controlar en este caso son el estado de la unidad RAID ("RAID Unit status") y el porcentaje completado de la operación que se esté llevando a cabo sobre el RAID en un momento dado ("RAID Unit percentage completed"). Las alertas se programan para avisar de cambios en el estado de la unidad RAID y para informar cuándo comienza y termina una determinada operación sobre la unidad (verificación, reconstrucción). Cuando concluya dicha operación se enviará una notificación a través del correo electrónico.
- **Eventos de información registrados en los Logs de Windows**: para los equipos que cuenten con Windows como sistema operativo (servidores de grabación y equipos de los centros de control), se informa convenientemente de los eventos recogidos por el log del sistema. Los ítems de Zabbix utilizados para estas alertas están agrupados en la sección "Windows Logs" de los servidores de grabación y de los equipos de los centros de control.

### 6.8.2. Nivel de "Advertencia" (Warning)

- **Enlace caído en un puerto de los conmutadores de la electrónica de red**: los ítems a controlar serán en este caso los correspondientes al estado administrativo y al estado operativo de la interfaz o puerto de un determinado

componente de la electrónica de red. Si el puerto está habilitado (el estado administrativo toma el valor "1") pero no está operativo (el estado operativo toma el valor "2"), se considerará que el enlace al puerto está caído.

- **Equipo sin conexión en la electrónica de red:** periódicamente se envía una solicitud de ping a los conmutadores de red monitorizados. En caso de que no respondan, se notificará a través de esta alerta.
- **Carga de CPU excesiva durante los últimos 3 minutos en el servidor de Zabbix**: si el ítem "CPU load" toma un valor superior a "2" en los últimos 180 segundos, se activará esta alerta.
- **Alta carga de procesador en el servidor Zabbix:** se activará esta alerta si el ítem "Processor Load" toma un valor superior a "5".
- **Excedido el 90% de uso de CPU en los equipos de los centros de control o en los servidores de grabación**: esta alerta se activará en caso de que el ítem "CPU usage" de los servidores de grabación o los equipos de los centros de control supere un valor del 90%.
- **Eventos de advertencia registrados en los Logs de Windows**: para los equipos que cuenten con Windows como sistema operativo (servidores de grabación y equipos de los centros de control), se informa convenientemente de los eventos recogidos por el log del sistema. Los ítems de Zabbix utilizados para estas alertas están agrupados en la sección "Windows Logs" de los servidores de grabación y de los equipos de los centros de control.
- **Cambio de versión del agente Zabbix instalado**: se comprobarán las modificaciones que sufra el ítem "Version of zabbix agent (d)".

### 6.8.3. Nivel "Medio" (Average)

- **Los demonios *syslogd, sshd, inetd y mysqld* no están ejecutándose en el servidor central Zabbix**: en este caso se comprobará si los ítems "Number of running processes inetd", "Number of running processes sshd", "Number of running processes syslogd" y "Number of running processes mysqld" toman el valor "0", en cuyo caso se activará la alerta.
- **Demasiados usuarios conectados al servidor central Zabbix:** en caso de que el ítem "Number of users connected" supere el valor "50" se activará la alerta.
- **Falta de memoria libre en el servidor central Zabbix:** si la memoria física libre disponible ("Free memory") es inferior a 10 MB, se activará la alerta.
- **Cambios en la configuración de los ficheros */usr/sbin/sshd, /usr/bin/ssh, /etc/services y /etc/passwd* en el servidor central Zabbix**: de forma periódica se hace un checksum de estos ficheros y, en caso de encontrarse diferencias, se activará la alerta. Los ítems involucrados son "Checksum of /usr/sbin/sshd",

"Checksum of /usr/bin/ssh", "Checksum of /etc/services" y "Checksum of /etc/passwd".

- **El servidor SSH no está ejecutándose en el servidor central Zabbix:** el ítem "SSH server is running" devuelve un valor asociado al estado del servidor SSH. En el momento en que dicho valor sea "0", se activará la alerta.
- **Falta de memoria libre en los equipos de los centros de control o en los servidores de grabación**: para valores del ítem "Free memory" inferiores a 10 MB se activará la alerta.
- **Reinicio de un servidor de grabación o de uno de los equipos de los centros de control:** si el valor correspondiente al ítem "Host uptime" en esos equipos es inferior a 600, la alerta se activará para avisar del reciente reinicio del servidor.
- **Cambios en la información de los equipos de los centros de control o en los servidores de grabación:** si se detectan cambios en el valor que toma el ítem "Host Information" en estos equipos, se activará la alerta.
- **Alta carga de procesador en los equipos de los centros de control o en los servidores de grabación:** se activará la alerta cuando el valor del ítem "Processor load" sea superior a "5".
- **Uso elevado de CPU en los equipos de los centros de control o en los servidores de grabación:** la alerta se activa cuando el porcentaje de uso de CPU dado por "CPU usage" sea superior al 90%.

### 6.8.4. Nivel "Alto" (High)

- **Cámara de videovigilancia sin conexión:** si el valor del ítem "Camera response" es igual a "0", se activará la alerta que indica que no hay conexión en la cámara.
- **Bajo nivel de espacio en disco en el servidor central Zabbix:** esta alerta se activa si el espacio disponible en disco dado por el ítem "Free disk space on /" es inferior a 10 MB.
- **El proceso *zabbix_server* no se está ejecutando en el servidor central Zabbix:** la alerta se activa cuando el ítem "Number of running processes zabbix_server" devuelve el valor "0".
- **El proceso *zabbix_agentd* no se está ejecutando en el servidor central Zabbix:** la alerta se activa cuando el ítem "Number of running processes zabbix_agentd" devuelve el valor "0".
- **El agente Zabbix (proceso *zabbix*_agentd) no se está ejecutando en los servidores de grabación o en los equipos de los centros de control:** al igual

que en el servidor Zabbix, esta alerta se activa cuando el ítem "Number of running zabbix agentd processes" toma un valor nulo.

- **El servidor *Apache* no se está ejecutando en el servidor central Zabbix:** la alerta se activará cuando el ítem "Number of running processes apache" tome un valor nulo.
- **El servidor HTTP no se está ejecutando en el servidor central Zabbix:** si el valor devuelto por el ítem "WEB (http) server is running" es nulo, se activará la alerta.
- **El servidor central Zabbix está caído (no responde a conexión):** si una solicitud de ping sobre el servidor no obtiene respuesta, se considerará que el servidor se encuentra sin conexión.
- **Demasiados procesos ejecutándose en el servidor central Zabbix**: si el número de procesos dado por el ítem "Number of processes" es superior a 300, se activará la alerta.
- **Demasiados procesos ejecutándose en los equipos de los centros de control o en los servidores de grabación**: de manera análoga al servidor Zabbix, si en estos equipos el ítem "Number of processes" supera el valor "300", se activará la alerta.
- **Fallo en uno de los discos del RAID de almacenamiento de los servidores de grabación**: existen varios valores asociados al estado de un disco (ver ANEXO IV. Mapeado de valores). En condiciones normales de funcionamiento, el valor que toma es "OK". En el momento en que ese valor, representado por el ítem que indica el estado de cada disco (Drive status) cambia y pasa a tomar uno de los posibles valores de error, se activará la alerta.
- **Temperatura de la CPU por encima de 60°C en los servidores de grabación**: se activará esta alerta en caso de que el valor dado por el ítem "CPU Temperature" sobrepase los 60°C.
- **Temperatura del sistema por encima de 60°C en los servidores de grabación**: se activará esta alerta en caso de que el valor dado por el ítem "System Temperature" sobrepase los 60°C.
- **Bajo nivel de espacio disponible (<10 MB) en disco en los equipos de los centros de control o en los servidores de grabación:** la alerta se activará cuando el espacio disponible en disco (Free disk space on C:) esté por debajo de los 10 MB.
- **Espacio libre en disco por debajo del 5% de la capacidad total en los servidores de grabación:** la alerta se activará cuando el espacio disponible en disco (Free disk space on C:) esté por debajo del 5%.

- **Uno de los servidores de grabación o uno de los equipos de los centros de control está caído (sin conexión):** si el valor devuelto por el ítem "Host status" es nulo, se activará la alerta.
- **El software de gestión *Sony RealShot Manager* no se está ejecutando en los servidores de grabación o en los equipos de los centros de control:** cuando el número de procesos en ejecución de este software es nulo, se activará la alerta.
- **Eventos de error registrados en los Logs de Seguridad, Sistema y Aplicación correspondientes a los equipos de los centros de control o a los servidores de grabación:** para los equipos que cuenten con Windows como sistema operativo (servidores de grabación y equipos de los centros de control), se informa convenientemente de los eventos recogidos por el log del sistema. Los ítems de Zabbix utilizados para estas alertas están agrupados en la sección "Windows Logs" de los servidores de grabación y de los equipos de los centros de control.

## 6.9. Envío de alertas por e-mail

De entre todas las alertas indicadas en el apartado anterior hay algunas para las que, además de notificar su activación a través del *frontend* de Zabbix, se asocia también el envío de un correo electrónico que avise del evento generado. Con ello "ahorraremos" tener que estar constantemente pendientes de la interfaz Web de Zabbix.

Para poder llevar a cabo envíos de correos electrónicos en Zabbix, primero es necesario configurar el mecanismo que permitirá a Zabbix comunicarse con una dirección de e-mail que nosotros le indiquemos. Accederemos a *Administration -> Media Types* y crearemos el medio para nuestra dirección de correo electrónico.

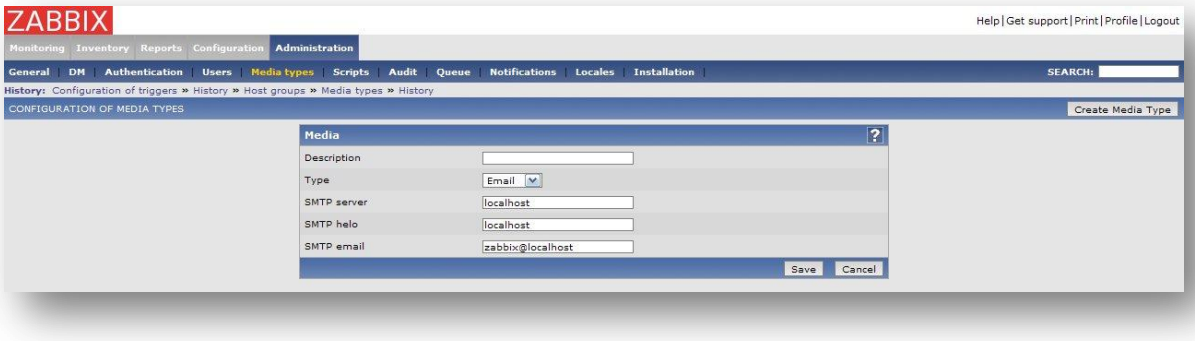



**Figura 32. Configuración de Zabbix para el envío de correos electrónicos**

Es importante que en el tipo especifiquemos que se trata de correo electrónico (tipo "Email") e indicaremos el servidor de correo electrónico y, finalmente, nuestra dirección.

Una vez configurada la dirección de correo electrónico, podremos enviar mensajes a ella informando de los eventos ocurridos en el sistema y lo haremos a través de las *Acciones* que Zabbix nos permite crear. Accedemos a *Configuration -> Actions* y seleccionamos como origen del evento a los disparadores o iniciadores que dan lugar a las alertas descritas en el apartado anterior.

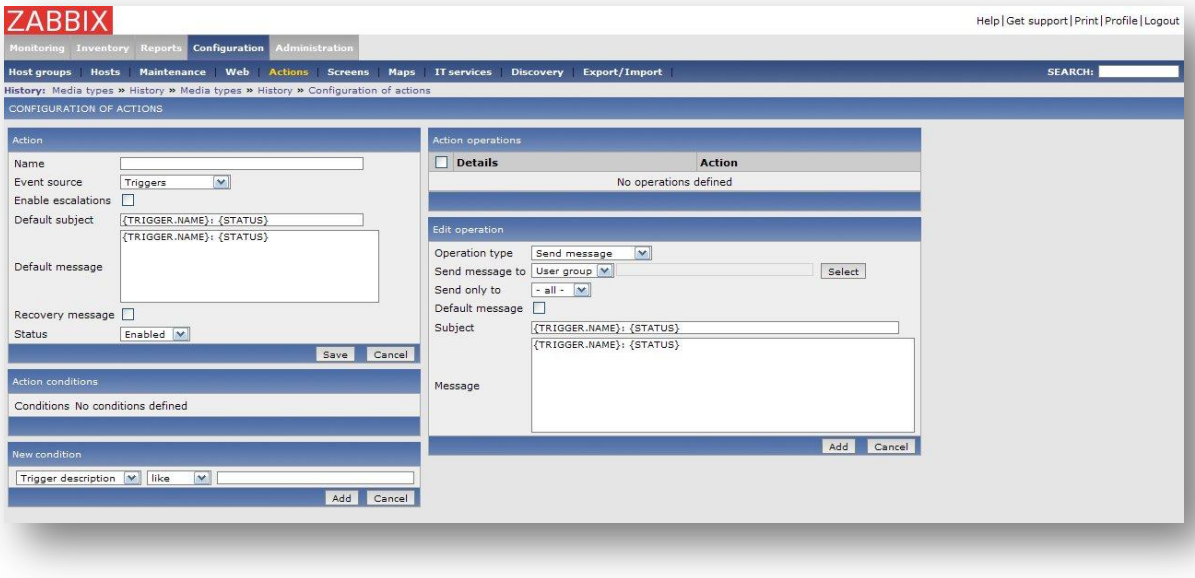



**Figura 33. Creación de acciones en Zabbix**

La idea es enviar un correo electrónico con las alertas que se consideran más importantes de cara a la disponibilidad del sistema de videovigilancia. El almacenamiento es uno de los aspectos más críticos y por ello se decide enviar un correo electrónico cada vez que se produzca alguna anomalía en ese aspecto. Además, Zabbix ofrece la posibilidad de enviar no sólo un correo electrónico cuando se produzca el problema, sino también cuando éste quede resuelto.

En resumen, enviaremos un correo electrónico cuando:

- **alguno de los discos de los sistemas RAID de almacenamiento presente fallo.**
- **el estado de la unidad RAID de alguno de los servidores de grabación pase a ser "DEGRADADO[20]".**

[20] La unidad se encuentra en este estado cuando uno o más de sus discos ha fallado.

- **la unidad RAID comienza un proceso de reconstrucción (ocurre cada vez que se reemplaza uno de sus discos).**
- **concluye el proceso de reconstrucción de una unidad RAID.**
- **el servidor Apache no está ejecutándose en el servidor Zabbix.**
- **una cámara se encuentra sin conexión.**
- **la temperatura de la CPU o del sistema está por encima de 60°C.**
- **el espacio en disco en los servidores de grabación es inferior al 5% (en este caso se enviará un correo de recuperación cuando el espacio vuelva a subir por encima del 5%).**
- **el espacio disponible en el servidor de Zabbix sea inferior a 10 MB.**
- **falle alguno de los 6 ventiladores de los servidores de grabación.**
- **Mysql no esté ejecutándose en el servidor Zabbix.**
- **se produzcan cambios en la configuración de los servidores de grabación o en los equipos de los centros de control.**
- **alguno de los servidores de grabación deje de tener conexión.**
- **el software Sony RealShot Manager no se esté ejecutando en los servidores de grabación o en los equipos de los centros de control.**
- **se reinicie el servidor Zabbix, uno de los servidores de grabación o uno de los equipos de los centros de control.**
- **no esté ejecutándose el agente Zabbix en el servidor Zabbix, en los servidores de grabación o en los equipos de los centros de control.**

Para redactar el asunto y el cuerpo del mensaje de correo electrónico que enviemos pueden utilizarse las macros que Zabbix implementa. Por ejemplo, *{TRIGGER.NAME}* nos indica el nombre del disparador que activa la alerta y *{STATUS}*, el estado de dicha alerta.

Como ejemplo, indicamos el formato del correo electrónico que se envía cuando se detecta que el porcentaje de espacio en disco está por debajo del 5%:

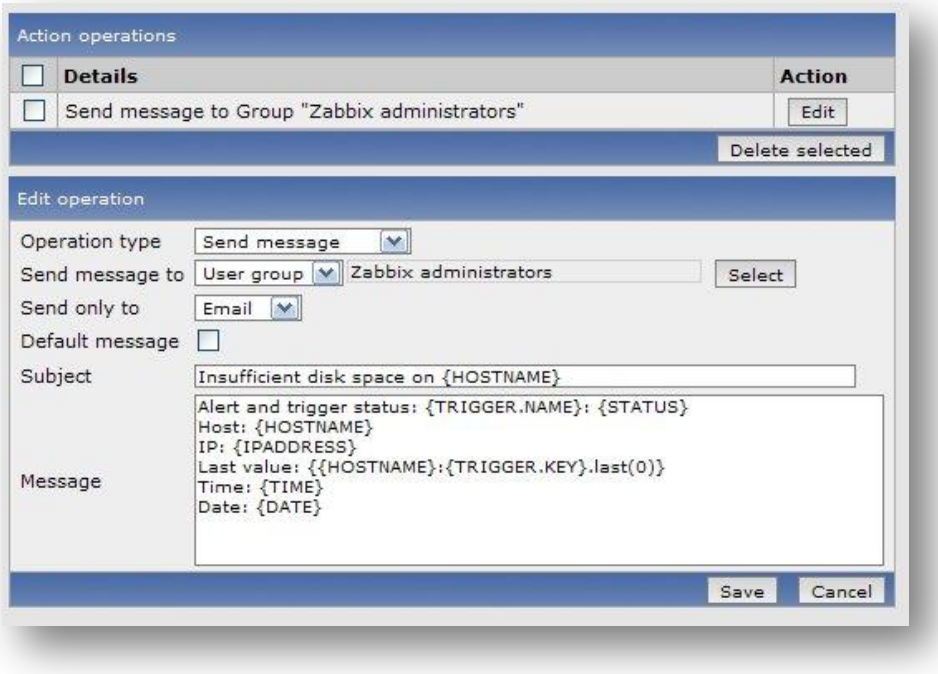
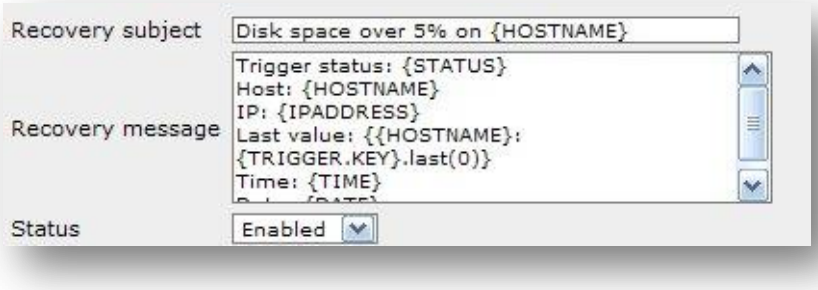



**Figura 34. Ejemplo de formato mensaje e-mail enviado por Zabbix**

En este caso concreto también enviamos un mensaje de recuperación (recovery message) para informar de que el problema ha quedado resuelto. Para el ejemplo del espacio en disco, este mensaje indica que dicho espacio está nuevamente por encima del 5% de la capacidad total.




Para más información sobre las macros que se pueden incluir en los mensajes de correo electrónico que Zabbix envía, ver ANEXO III. Macros implementadas en Zabbix.

## 6.10. Mapas y gráficos

A la hora de interpretar los datos de monitorización suelen ser de gran ayuda los gráficos, pues nos permiten no sólo conocer el valor actual de un cierto parámetro sino también su evolución en el tiempo. Para crear un gráfico en Zabbix, accederemos al *host* o equipo al cual queramos asignar el gráfico desde *Configuration -> Hosts -> Create graph*.

Los gráficos se han creado especialmente para los servidores de grabación y para el servidor central Zabbix y muestran los siguientes parámetros de monitorización:

**❶ Servidores de grabación**

- Temperatura de la CPU.
- Temperatura del sistema.
- Utilización del espacio en disco.
- Utilización de CPU.
- Memoria libre.
- Tráfico de red.

**❷ Servidor central Zabbix**

- Utilización de CPU.
- Utilización del espacio en disco.
- Número de *ítems* en la base de datos.
- Número de nuevos valores por segundo.
- Número de valores en la tabla del historial.
- Tráfico de red en sus dos interfaces (1 y 2).

Las opciones que se nos ofrecen a la hora de crear los grafos son variadas y permiten incluir varios parámetros de monitorización cuyos valores se dibujarán en el gráfico. Para un mayor detalle de los gráficos creados, ver ANEXO VI. Gráficos de monitorización.

También es posible crear mapas con los que representar el estado de los equipos monitorizados. Para crearlos, iremos al menú *Configuration -> Maps*. Actualmente se han creado mapas en los que se muestra el estado de los servidores, incluyendo los problemas que pudieran tener, y mapas en los que se puede observar la ubicación de cada cámara de videovigilancia en los edificios de la UC3M.

## 6.11. Comandos remotos

Las acciones programadas en Zabbix (*Configuration -> Actions*) pueden tener asociadas, además del envío de un mensaje, la ejecución de comandos remotos **[43]**.

Estos comandos se crean estableciendo los siguientes parámetros al crear un acción en Zabbix:

| Parámetro | Descripción |
|---|---|
| **Tipo de acción** | 'Remote command' |
| **Comando remoto** | Cada línea debe contener un comando que será ejecutado remotamente. |

**Tabla 49. Parámetros de creación de una acción con comando remoto en Zabbix**

La sintaxis a seguir para definir el comando remoto es la siguiente:

| COMANDO REMOTO | Descripción |
|---|---|
| **<host>:<command>** | El comando 'command' se ejecutará en el host 'host'. |
| **<group>#<command>** | El comando 'command' se ejecutará en todos los hosts del grupo 'group'. |

**Tabla 50. Sintaxis de creación de un comando remoto en Zabbix**

En la plataforma de monitorización actual se han implementado comandos remotos para el reinicio del servidor Apache en el servidor central de Zabbix y para el reinicio de la aplicación *Sony RealShot Manager* cuando se detecta que no está ejecutándose en alguno de los servidores de grabación.

## 6.12. User Parameters

Las posibilidades que Zabbix ofrece con sus claves de monitorización por defecto pueden extenderse con la definición de parámetros personalizados por el usuario. Para ello se debe editar el fichero de configuración del agente Zabbix instalado y seguir la siguiente sintaxis:

```
UserParameter=key,command
```

Donde,

| Parámetro | Descripción |
|---|---|
| **Key** | Clave del item |
| **Command** | Comando a ejecutarse para obtener el valor de la clave. |

**Tabla 51. Definición de parámetros de usuario (User parameters) en Zabbix**

Cuando deseemos monitorizar un parámetro de monitorización que haga uso de un *User Parameter*, crearemos un *ítem* como otro cualquiera indicando el valor del dato *key* que hayamos indicado en el *User Parameter* definido.

Las ventajas de esta técnica son obvias, y es que permiten ejecutar cualquier comando siempre que éste devuelva algún tipo de valor. En contrapartida, sólo pueden configurarse para equipos que tengan un agente Zabbix instalado, en cuyo caso se debe editar manualmente el fichero de configuración correspondiente.

# 7

# Plan de pruebas

# 7. PLAN DE PRUEBAS

Se describe aquí el plan de pruebas confeccionado con objeto de verificar que el resultado final obtenido al crear el sistema de monitorización se adecúa a las necesidades iniciales descritas en la fase de análisis.

*Métrica v3* [10] define una serie de pruebas a desarrollar a distintos niveles:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de sistema.
- Pruebas de implementación.
- Pruebas de aceptación (usabilidad).

El plan de pruebas diseñado para la plataforma de monitorización incluirá pruebas unitarias, pruebas del sistema y pruebas de aceptación.

Todas esas pruebas se escribirán en forma de tabla con el siguiente formato:

| **IDENTIFICADOR:** | |
|---|---|
| **Propósito** | |
| **Pasos a ejecutar** | |
| **Salida o estado esperado** | |

El identificador se define siguiendo la línea marcada en la especificación de requisitos de la plataforma de monitorización:


Donde *TipoPrueba* tomará el valor "PU" si se trata de una prueba unitaria, "PI" si es una prueba de integración o "PA" si se refiere a una prueba de aceptación.

## 7.1. Pruebas unitarias

Citando textualmente a Métrica V3, las pruebas unitarias "*son las pruebas que comprenden las verificaciones asociadas a cada componente del sistema de información. Su realización tiene como objetivo verificar la funcionalidad y estructura de cada componente individualmente*".

En este caso, las pruebas unitarias planificadas son del tipo que Métrica V3 denomina **"de caja negra o enfoque funcional"**. Se comprueba que cada componente cumpla con su funcionalidad sin deternos en su estructura interna.

De lo que se trata con estas pruebas es de verificar que los componentes descritos en la arquitectura de la plataforma de monitorización funcionan correctamente de forma independiente.

| **IDENTIFICADOR: CCTV-MON-PU-001** | |
|---|---|
| **Propósito** | Comprobación de los agentes Zabbix |
| **Pasos a ejecutar** | En los sistemas Windows, comprobaremos que existe un servicio para el agente Zabbix y que dicho servicio está arrancado. Para probar si los agentes aceptan conexiones en el puerto 10050, ejecutaremos localmente un *telnet* a dicho puerto. |
| **Salida o estado esperado** | En la lista de servicios del sistema aparecerá un servicio por nombre 'Zabbix Agent' y con estado 'Iniciado'. La ejecución del telnet no mostrará ningún mensaje de error. |

**Tabla 52. Prueba CCTV-MON-PU-001**

| **IDENTIFICADOR: CCTV-MON-PU-002** | |
|---|---|
| **Propósito** | Comprobación de los agentes SNMP |
| **Pasos a ejecutar** | Para llevar a cabo esta prueba ejecutaremos el comando *snmpwalk* sobre cada equipo de la plataforma que tenga activado un agente SNMP. Ejecutaremos además un *telnet* al puerto 161 de cada uno de esos equipos. |
| **Salida o estado esperado** | Tras ejecutar el comando, se deberá mostrar un listado con valores obtenidos a través de SNMP. Además, en los sistemas Windows, dentro de la lista de servicios del aparecerá un servicio para el agente SNMP y con estado 'Iniciado'. La ejecución del telnet no mostrará ningún mensaje de error. |

**Tabla 53. Prueba CCTV-MON-PU-002**

| IDENTIFICADOR: CCTV-MON-PU-003 | |
|---|---|
| **Propósito** | Comprobación del servidor central Zabbix |
| **Pasos a ejecutar** | El servidor central deberá aceptar conexiones en el puerto 10050 (agente Zabbix) y en el puerto 10051 (envío de ***active checks***). Ejecutaremos un *telnet* a ambos puertos. |
| **Salida o estado esperado** | La ejecución del telnet no mostrará ningún mensaje de error. |

**Tabla 54. Prueba CCTV-MON-PU-003**

| IDENTIFICADOR: CCTV-MON-PU-004 | |
|---|---|
| **Propósito** | Comprobación del servidor Apache |
| **Pasos a ejecutar** | Desde la máquina "zabbix-cctv" ejecutaremos un navegador web y en la barra de direcciones escribiremos *http://localhost* o bien *http://127.0.0.1* |
| **Salida o estado esperado** | Se nos mostrará en el navegador un mensaje indicando que el servidor Apache funciona. |

**Tabla 55. Prueba CCTV-MON-PU-004**

| IDENTIFICADOR: CCTV-MON-PU-005 | |
|---|---|
| **Propósito** | Comprobación del funcionamiento de PHP |
| **Pasos a ejecutar** | Crearemos un archivo de test en PHP con el contenido <?php phpinfo(); ?> y por nombre "test.php" en la máquina "zabbix-cctv". Ejecutaremos un navegador web en esa máquina y en la barra de direcciones escribiremos *http://localhost/test.php*. |
| **Salida o estado esperado** | Se nos mostrará información sobre la versión PHP instalada. |

**Tabla 56. Prueba CCTV-MON-PU-005**

| IDENTIFICADOR: CCTV-MON-PU-006 | |
|---|---|
| **Propósito** | Comprobación del funcionamiento de MySQL |
| **Pasos a ejecutar** | Una vez hayamos comprobado que el servidor Apache y PHP funcionan correctamente, accederemos a la interfaz web de Zabbix en la plataforma de monitorización. |
| **Salida o estado esperado** | En caso de fallo aparecerán errores de conexión a la base de datos MySQL. De otro modo, la interfaz web no mostrará ningún problema. |

**Tabla 57. Prueba CCTV-MON-PU-006**

## 7.2. Pruebas de integración

En referencia a las pruebas de integración, Métrica V3 indica que "*las pruebas de integración comprenden verificaciones asociadas a grupos de componente, generalmente reflejados en la definición de subsistemas de construcción o en el plan de integración del sistema de información. Tienen por objetivo verificar el correcto ensamblaje entre los distintos componentes*".

Las pruebas de integración llevadas a cabo se clasifican como pruebas de **comunicación** entre los distintos componentes de la plataforma de monitorización..

### 7.2.1. Pruebas de comunicación

Verificaremos en este punto la comunicación que existe entre el servidor central de Zabbix y los demás componentes de la plataforma: servidores de grabación, equipos de los centros de control, cámaras de videovigilancia y equipos de la electrónica de red.

| **IDENTIFICADOR: CCTV-MON-PI-001** | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y los servidores de grabación (vía agente Zabbix) |
| **Pasos a ejecutar** | Una de las vías de comunicación entre el servidor Zabbix y los servidores de grabación es a través del agente Zabbix instalado en éstos. Se comprobará que los agentes aceptan peticiones del servidor central en el puerto 10050, tal como se define en sus ficheros de configuración. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix que los servidores de grabación están marcados como "*Monitored*" seguido de un icono indicativo en color verde. |

**Tabla 58. Prueba CCTV-MON-PI-001**

| IDENTIFICADOR: CCTV-MON-PI-002 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y los equipos de los centros de control (vía agente Zabbix) |
| **Pasos a ejecutar** | Al igual que en los servidores de grabación, los equipos de los centros de control se comunican con el servidor central a través de agentes Zabbix instalados en ellos. Nuevamente, verificaremos que esos agentes aceptan peticiones del servidor central en el puerto 10050. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix que estos equipos están marcados como "*Monitored*" seguido de un icono indicativo de color verde. |

**Tabla 59. Prueba CCTV-MON-PI-002**

| IDENTIFICADOR: CCTV-MON-PI-003 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y los servidores de grabación (vía SNMP) |
| **Pasos a ejecutar** | Los servidores de grabación también pueden comunicarse con el servidor Zabbix por medio de un agente SNMP instalado en ellos. Por defecto, el puerto en el que los servidores de grabación deberán escuchar peticiones a paquetes SNMP es el 161. Lo verificaremos ejecutando la funcionalidad *snmpwalk* desde el servidor Zabbix o bien comprobando que dicho puerto está abierto. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix que estos equipos están marcados como "*Monitored*" con un icono indicativo de color verde correspondiente a SNMP. |

**Tabla 60. Prueba CCTV-MON-PI-003**

| IDENTIFICADOR: CCTV-MON-PI-004 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y equipos de los centros de control (vía SNMP) |
| **Pasos a ejecutar** | Los equipos de los centros de control también pueden comunicarse con el servidor Zabbix por medio de un agente SNMP instalado en ellos. Por defecto, el puerto en el que los servidores de grabación deberán escuchar peticiones a paquetes SNMP es el 161. Lo verificaremos ejecutando la funcionalidad *snmpwalk* desde el servidor Zabbix o bien comprobando que dicho puerto está abierto. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix que estos equipos están marcados como "*Monitored*" con un icono indicativo de color verde correspondiente a SNMP. |

**Tabla 61. Prueba CCTV-MON-PI-004**

| IDENTIFICADOR: CCTV-MON-PI-005 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y las cámaras de videovigilancia (vía SNMP) |
| **Pasos a ejecutar** | Las cámaras no permiten instalar un agente Zabbix en ellas, pero sí es posible habilitar un agente SNMP. Al igual que los servidores de grabación y los equipos de los centros de control, el puerto utilizado por defecto para que ese agente SNMP escuche peticiones del servidor Zabbix es el 161. Lo verificaremos ejecutando la funcionalidad *snmpwalk* desde el servidor Zabbix o bien comprobando que dicho puerto está abierto. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix que las cámaras están marcadas como "*Monitored*" con un icono indicativo de color verde correspondiente a SNMP. |

**Tabla 62. Prueba CCTV-MON-PI-005**

| IDENTIFICADOR: CCTV-MON-PI-006 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y equipos de electrónica de red (vía SNMP) |
| **Pasos a ejecutar** | Al igual que en las cámaras, en estos equipos no es posible instalar un agente Zabbix, pero sí un agente SNMP. Como en otros casos, el puerto utilizado por defecto para que ese agente SNMP escuche peticiones del servidor Zabbix es el 161. Lo verificaremos ejecutando la funcionalidad *snmpwalk* desde el servidor Zabbix o bien comprobando que dicho puerto está abierto. |
| **Salida o estado esperado** | De no haber ningún problema, observaremos en Zabbix estos equipos están marcados como "*Monitored*" con un icono indicativo de color verde correspondiente a SNMP. |

**Tabla 63. Prueba CCTV-MON-PI-006**

| IDENTIFICADOR: CCTV-MON-PI-006 | |
|---|---|
| **Propósito** | Comunicación entre el servidor Zabbix y los agentes Zabbix de la plataforma para el envío de active checks |
| **Pasos a ejecutar** | Comprobaremos que el servidor central Zabbix acepta conexiones en el puerto 10051 (active checks) y para ello crearemos un *item* en Zabbix de tipo *active*. |
| **Salida o estado esperado** | En caso de que el puerto 10051 acepte conexiones, se observará cómo el *item* creado va tomando valores. |

**Tabla 64. Prueba CCTV-MON-PI-006**

## 7.3. Pruebas de aceptación

Según Métrica V3, "*las pruebas de aceptación van dirigidas a validar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir la aceptación final por parte del usuario*".

### 7.3.1. Pruebas funcionales

Las pruebas del sistema de tipo funcional se reservan para probar que el sistema cumple con las funcionalidades especificadas en los requisitos.

Básicamente se trata de diseñar pruebas con las que comprobar si la plataforma de monitorización desplegada cubre los requisitos especificados en la fase de análisis.

Incluiremos aquí las pruebas definidas para comprobar que se satisfacen los requisitos de funcionamiento más representativos.

| **IDENTIFICADOR: CCTV-MON-PA-001** | |
|---|---|
| **Propósito** | Comprobación de la inserción de equipos |
| **Pasos a ejecutar** | Introduciremos un equipo en la plataforma a través del frontend de Zabbix. |
| **Salida o estado esperado** | En el frontend veremos que el equipo ha quedado monitorizado y nos conectaremos a la base de datos MySQL para ejecutar una consulta con la que comprobar que el equipo ha sido efectivamente creado e insertado. |

**Tabla 65. Prueba CCTV-MON-PA-001**

| **IDENTIFICADOR: CCTV-MON-PA-002** | |
|---|---|
| **Propósito** | Comprobación de la inserción parámetros de monitorización |
| **Pasos a ejecutar** | Crearemos un *item* en Zabbix dentro de una plantilla y vincularemos ésta con un equipo ya introducido en la plataforma. |
| **Salida o estado esperado** | El parámetro o *item* creado tomará valores para el equipo al cual se ha asignado. |

**Tabla 66. Prueba CCTV-MON-PA-002**

| **IDENTIFICADOR: CCTV-MON-PA-003** | |
|---|---|
| **Propósito** | Comprobación de las reglas de descubrimiento |
| **Pasos a ejecutar** | Crearemos una regla de descubrimiento dentro del rango de direcciones IP de la red privada CCTV. A continuación, conectaremos físicamente un nuevo equipo a esa red y le asignaremos una dirección IP dentro del rango definido para la regla de descubrimiento creada. |
| **Salida o estado esperado** | Zabbix detectará ese nuevo equipo con su dirección IP y lo insertará automáticamente en la plataforma de monitorización. |

**Tabla 67. Prueba CCTV-MON-PA-003**

| IDENTIFICADOR: CCTV-MON-PA-004 | |
|---|---|
| **Propósito** | Comprobación de las alertas de notificación |
| **Pasos a ejecutar** | Para esta prueba utilizaremos un disparador concreto que hayamos creado, como por ejemplo, el indicador de espacio en disco por debajo del 5%. Dicho disparador tiene asociada la acción de envío de un correo electrónico para notificar al administrador de la plataforma de monitorización.<br>Ajustaremos el umbral del disparador a un valor superior al porcentaje de espacio libre en disco en el momento actual para así inducir a su activación y a la ejecución de la acción que envía el correo electrónico.<br>Procederemos de manera similar al resto de alertas de notificación programadas. |
| **Salida o estado esperado** | El administrador de la plataforma recibirá un mensaje en su bandeja de correo electrónico informando del evento. |

**Tabla 68. Prueba CCTV-MON-PA-004**

| IDENTIFICADOR: CCTV-MON-PS-005 | |
|---|---|
| **Propósito** | Comprobación de comandos remotos |
| **Pasos a ejecutar** | Los comandos remotos se crean dentro de una acción asociada a la activación de un disparador. Reproduciremos las condiciones que activan ese disparador para que los comandos remotos definidos se ejecuten. |
| **Salida o estado esperado** | Se observará un efecto u otro dependiendo del comando remoto ejecutado. Por ejemplo, si detenemos el servidor Apache voluntariamente, Zabbix ejecutará un comando remoto que lo reinicie de nuevo. |

**Tabla 69. Prueba CCTV-MON-PA-005**

| IDENTIFICADOR: CCTV-MON-PS-006 | |
|---|---|
| **Propósito** | Interfaz Web para interacción del usuario |
| **Pasos a ejecutar** | Se ejecutarán varias operaciones a través del interfaz Web para observar posibles dificultades que pueda plantear al usuario. Se evaluará el uso de imágenes y de elementos de navegación y orientación al usuario. |
| **Salida o estado esperado** | La interfaz deberá resultar intuitiva y "amigable" desde el punto de vista de la usabilidad sin plantear dificultades navegacionales. |

**Tabla 70. Prueba CCTV-MON-PS-006**

# 8

# Plan de mantenimiento

# 8. PLAN DE MANTENIMIENTO

En los sistemas informáticos es habitual definir un mantenimiento en el que se lleven a cabo una serie de acciones dirigidas a garantizar que el sistema funcione correctamente, ya sea desde el punto de vista hardware o software.

Ese mantenimiento se puede establecer a varios niveles. En este caso consideraremos un mantenimiento a niveles **preventivo**, **correctivo** y **evolutivo**, definidos por el estándar Métrica V3 [10]. El mantenimiento **preventivo** está orientado a *encontrar y corregir los problemas antes de que éstos produzcan fallos*. La idea es evitar el mayor número de incidencias en el sistema, como son aquellas que puedan suponer reparaciones en el sistema o que comprometan su seguridad, su configuración o la integridad de los datos con los que trabaja. Por su parte, el mantenimiento **correctivo** se encamina a corregir los problemas del sistema una vez ya se han producido. Finalmente, en el mantenimiento **evolutivo** se realizan las incorporaciones y/o modificaciones necesarias para cubrir nuevas necesidades.

A continuación mostraremos las operaciones de mantenimiento aplicadas sobre la plataforma de monitorización del sistema de videovigilancia de la UC3M.

## 8.1. Mantenimiento preventivo

Las políticas actuales seguidas en la línea del mantenimiento preventivo se centran en la base de datos MySQL con la cual trabaja Zabbix. Las tareas básicas que, tradicionalmente, se llevan a cabo en el mantenimiento de una base de datos MySQL son las siguientes:

- **Limpieza y rotación de Logs:** los ficheros de log de MySQL pueden alcanzar grandes tamaños que ocasionan una disminución del espacio en disco y pueden llegar a ralentizar el sistema. Limpiar esos registros periódicamente y rotarlos según un tiempo determinado ayuda a evitar su crecimiento descontrolado.
- **Optimización de tablas:** las tablas de la base de datos MySQL son, al fin y al cabo, archivos. Como tales, tras varias operaciones de actualización de datos, pueden ocasionar problemas de fragmentación, pueden ocupar espacio inútil ya que éste no se reasigna automáticamente al eliminar registros, etc. Por ello se recomienda optimizar periódicamente dichas tablas. No obstante, se debe tener en cuenta que cuanto mayor sea el tamaño de la tabla, mayor tiempo supondrá la optimización de la misma.
- **Vaciar caché de consultas:** los resultados de las consultas son almacenados en la caché de consultas, en la cual también es posible observar fragmentación,

sobre todo si se ejecutan sentencias *SELECT* con resultados de diferente tamaño. Para evitar estos problemas, es recomendable vaciar frecuentemente esta caché de consultas.

Por otra parte, para optimizar el rendimiento de la base de datos se llevan a cabo ajustes (***tuning***) sobre diferentes parámetros de su configuración.

### 8.1.1. Limpieza y rotación de Logs de MySQL

MySQL dispone de varios archivos de log en los que se registran las operaciones sobre la base de datos de Zabbix.

| Archivo de registro | Tipo de información registrada en el archivo |
|---|---|
| **Registro de error** | Registra problemas encontrados iniciando, ejecutando, o parando **mysqld**. |
| **Registro de consultas** | Registra las conexiones de clientes establecidas, y las sentencias ejecutadas. |
| **Registro de actualizaciones** | Registra las sentencias que cambian datos. Este registro está ya en desuso. |
| **Registro binario** | Registra todas las sentencias que cambian datos. También utilizado para replicación. |
| **Registro de lentitud** | Registra todas las sentencias que tardaron más de `long_query_time`[21] segundos en ejecutarse, o no utilizaron índices. |

**Tabla 71. Ficheros de registro de MySQL**

El tamaño de estos ficheros de log crece considerablemente a medida que aumenta la carga sobre la base de datos, algo que con frecuencia ocurre cuando se incrementa el número de equipos y parámetros monitorizados por Zabbix.

Si queremos evitar tener Logs de MySQL que sean intratables por su tamaño tendremos que llevar a cabo operaciones de mantenimiento sobre los mismos.

MySQL crea un nuevo archivo de log binario cada vez que es reiniciado o cada vez que el tamaño de dicho archivo llegue a un tamaño máximo fijado. Además de ese tamaño, se puede establecer durante cuánto tiempo se mantendrán. Esos dos parámetros (tamaño y persistencia en el tiempo) se personalizan editando las opciones de configuración correspondientes (*max_binlog_size* y *expire_logs_days*) en el fichero "*my.cnf*"[22]. Actualmente, dichos parámetros están configurados con los valores **50MB** y **7 días**, respectivamente.

[21] Parámetro de configuración de MySQL para las consultas lentas
[22] Fichero de configuración de MySQL

Para forzar a MySQL a que comience a utilizar archivos de registro nuevos y que, de esta manera, se limpien, ejecutaremos la sentencia "*FLUSH LOGS*". En el caso de la base de datos de Zabbix, combinaremos la limpieza con la rotación de los archivos de registro.

La rotación de archivos de log se lleva a cabo a través de "*logrotate*", utilidad que nos permitirá gestionar de manera adecuada esos archivos para que su tamaño no suponga un obstáculo en el rendimiento del sistema.

Habitualmente, *logrotate* se ejecuta como un trabajo programado en el *crontab*[23] del sistema y lo hace leyendo las opciones de un archivo de configuración específico. Ese archivo es un script en el que se encuentran las opciones y operaciones a efectuar para el tratamiento de los ficheros de log de mysql. El contenido de este script puede localizarse en ANEXO XI. Script de rotación de logs MySQL.

### 8.1.2. Optimización de tablas MySQL

Para optimizar las tablas de la base de datos MySQL de Zabbix se ejecuta el comando *OPTIMIZE TABLE nombre_de_tabla*. La ejecución de este comando desfragmentará la tabla o tablas que le indiquemos y es muy útil en el caso de que éstas sufran frecuentes operaciones de actualización.

Es recomendable ejecutar regularmente este comando sobre las tablas que sufren constantes borrados de registros aunque, como precaución, no debe olvidarse que la tabla queda bloqueada mientras está siendo optimizada, por lo que deberemos prestar especial atención en el caso de tablas grandes.

### 8.1.3. Vaciar caché de consultas

La caché de consultas de MySQL puede sufrir los mismos problemas de fragmentación que las tablas de datos debido a diferentes tamaños en los resultados de las consultas *SELECT*. Para vaciar la caché de consultas ejecutaremos periódicamente el comando ***FLUSH QUERY CACHE***.

### 8.1.4. Ajustes de rendimiento sobre la base de datos Zabbix

Con objeto de mejorar el rendimiento de la base de datos MySQL de Zabbix, es posible ajustar parámetros de su configuración (*tuning*) y así evitar problemas tales

[23] Fichero que contiene los procesos que deben ejecutarse de forma regular según el administrador de procesos "cron" de los sistemas Linux.

como la lentitud en las consultas que se hagan sobre la base de datos. Asimismo, otro de los factores a tener en cuenta es el espacio, ya que, a medida que avanza el tiempo, las tablas de la base de datos pueden crecer de forma considerable si no se controla de alguna manera.

En Zabbix se deben vigilar especialmente estos aspectos. Si la base de datos no es capaz de responder de forma eficiente a las constantes actualizaciones sobre los datos de monitorización, es muy posible que gran parte de esos datos queden encolados (ver menú *Adminstration -> Queue* en Zabbix) y no lleguen a mostrarse en tiempo real tal como debieran. Si el número de registros almacenados en las tablas se vuelve intratable, tarde o temprano nos quedaremos sin espacio físico donde almacenarlos y, además, ralentizaremos las operaciones sobre la base de datos.

Las consideraciones más importantes que debemos tener en cuenta a la hora de optimizar el funcionamiento de la base de datos de Zabbix se centran en los *ítems* o parámetros de monitorización y en los disparadores creados:

**❶ Items**

- **Adoptar una actitud realista:**
  - Lo primero que debemos vigilar es el intervalo de actualización de cada ítem. Intervalos de 5 a 60 segundos supondrán un incremento en la carga sobre la base de datos, por lo que utilizaremos un intervalo mayor, especialmente en el caso de aquellos parámetros de monitorización que no sufran cambios en el tiempo de forma frecuente.
  - Se debe tener un cierto control sobre el historial de los datos, evaluando el tiempo que mantenemos éstos en el sistema y el espacio que ello supone. En su lugar, recurriremos a los "*datos de tendencias*" o *trends*, ya que toma los valores del historial pero los almacena con un intervalo más largo.

**❷ Disparadores**

- Cada disparador genera un registro en la base de datos asociado al evento que lo activa. La tabla que almacena estos eventos puede crecer de forma significativa si existen disparadores sin utilidad alguna o constantemente cambiando de estado. Para evitarlo, crearemos disparadores que nos resulten realmente útiles y cuyas condiciones estén afinadas en la medida de lo posible.

**❸ Control de la caducidad de eventos y acciones (Housekeeping)**

- La propia herramienta Zabbix puede ejecutar periódicamente una "limpieza" de las acciones y eventos con una antigüedad superior a la que establezcamos en las opciones del frontend (*Administration -> General -> Entrada "Housekeeping" del menú desplegable*). Sin embargo, esto no supone la eliminación de los registros que contengan esos datos, por lo que es posible que queden registros huérfanos o vacíos. En cualquier caso, la limpieza de esos eventos significará que no se activarán disparadores de forma innecesaria y, por ende, que no se ejecutarán las acciones que éstos tengan asociadas.

Son varios los factores que pueden limitar el rendimiento de una base de datos MySQL, tanto a nivel hardware como a nivel software. Elementos hardware como los discos utilizados para el almacenamiento o la memoria física disponible, si no ofrecen suficientes prestaciones, pueden empeorar el funcionamiento de la base de datos MySQL.

Pero, como decíamos, también hay parámetros software que deben tenerse en cuenta de cara a mejorar el rendimiento de la base de datos. El primero de ellos lo consitituyen los **búferes y caché de MySQL**. La mayoría de la memoria reservada por MySQL se utiliza para varios búferes y caché internos. Estos búferes se encuentran en dos grandes grupos: *búferes globales* y *para conexiones*.

Antes de tratar sobre esos búferes, aclararemos que el motor de almacenamiento utilizado para la base de datos actual de Zabbix es ***MyISAM*** **[13]**. Básicamente, este motor de almacenamiento crea, por cada tabla, tres archivos. Uno de ellos para los datos (con extensión ".MYD"), otro para los índices (extensión ".MYI") y un último fichero para almacenar la definición de los datos de la tabla (extensión ".frm").

Volviendo a los búferes de MySQL, uno de los búferes más importantes es el búfer de claves MyISAM (*key_buffer_size*). El búfer de claves de MyISAM es el lugar donde MySQL almacena en caché los bloques utilizados más frecuentemente por los datos de índices de las tablas. Cuanto menos necesite acceder MySQL a disco para los índices de tabla, más rápidas serán las consultas. Por ello, y en la medida de lo posible, se debe considerar la posibilidad de hacer que el búfer de claves sea lo suficientemente grande para contener los índices de las tablas más utilizadas. Observando el tamaño de los archivos .MYI de las tablas podemos hacernos una idea de cómo de grande debe ser el valor del tamaño del búfer de claves.

Sin embargo, son dos las limitaciones que encontramos a la hora de establecer el tamaño de ese búfer de claves. No podemos guiarnos por el tamaño de los archivos .MYI de las tablas más utilizadas porque, en algunos casos, dicho tamaño sobrepasa la memoria física disponible. En segundo lugar, debemos parametrizar ese búfer de

manera que el mínimo de memoria necesaria no supere la memoria física disponible. De lo contrario, probablemente nos encontremos con un cuello de botella en la memoria.

Para el cálculo de la memoria mínima necesaria se utiliza la siguiente expresión:

**min_memoria_necesaria = búferes globales + (búferes_por_trama * max_conexiones)**

donde los búferes por trama incluyen:

**sort_buffer**
**myisam_sort_buffer**
**read_buffer**
**join_buffer**
**read_rnd_buffer**

y los búferes globales incluyen:

**key_buffer**
**innodb_buffer_pool**
**innodb_log_buffer**
**innodb_additional_mem_pool**
**net_buffer**

Una manera de determinar si el valor asignado al búfer de claves de MyISAM es adecuado es utilizar una herramienta de monitorización de MySQL, como, por ejemplo, la aplicación *mytop* **[14]**. Con esta aplicación podemos observar, entre otros parámetros de rendimiento, la eficiencia de las claves MyISAM. En función del porcentaje de eficiencia que obtengamos podremos determinar si el valor que asignamos al tamaño del búfer de claves es óptimo.

```
MySQL on localhost (5.0.75-oubuntu10-log)                    up 0+00:42:50
[16:37:22] Queries: 973.0  qps:   0 Slow:    0.0       Se/In/Up/De(%):   741/00/00/00          qps now:   0
Slow qps: 0.0  Threads:  15 (  2/  8) 00/00/00/00  Key Efficiency: 99.4%  Bps in/out:  8.0/ 1.5k
Now in/out:  8.3/ 1.4k

Id      User    Host/IP          DB       Time    Cmd     Query or State
--      ------  ----------       ---      --------------  ----------------------
119     zabbix  localhost        zabbix   0       Query   insert into history_uint (itemid,clock,value)
values (25274,1282055835,42, ...)
120     root    localhost        zabbix   0       Query   show full process list
116     zabbix  localhost        zabbix   3       Sleep
124     zabbix  localhost        zabbix   4       Sleep
117     zabbix  localhost        zabbix   5       Sleep
118     zabbix  localhost        zabbix   6       Sleep
121     zabbix  localhost        zabbix   22      Sleep
122     zabbix  localhost        zabbix   22      Sleep
137     zabbix  localhost        zabbix   23      Sleep
```

**Figura 35. Salida del commando mytop**

En caso de encontrarnos con problemas a nivel de memoria, otras opciones que podemos considerar para restablecer el balance entre ésta y la memoria que necesita MySQL son las siguientes:

- Añadir más memoria.
- Reducir el número de conexiones máximas.
- Reducir el tamaño de algunos búferes por trama.

## 8.2. Mantenimientos correctivo y evolutivo

Conceptualmente, en el mantenimiento **correctivo** se corrigen errores o aquellos aspectos del sistema que no están funcionando como debieran, mientras que el mantenimiento **evolutivo** se centra en las modificaciones necesarias en el sistema para ajustarse a procesos de expansión o a nuevas necesidades por parte de los usuarios del mismo.

Un ejemplo de una operación que puede encuadrarse tanto en el mantenimiento correctivo como en el evolutivo son las sucesivas actualizaciones que ha sufrido la herramienta Zabbix a lo largo del desarrollo de este proyecto, pues, con cada actualización se corrigen errores como pueden ser vulnerabilidades de seguridad y a la vez se cubren nuevas necesidades surgidas de los procesos de revisión a los que se somete por parte de los creadores de Zabbix como de sus usuarios.

En la siguiente tabla mostraremos las actualizaciones llevadas a cabo en Zabbix (incluyendo su correspondiente versión) y los motivos que impulsaron a ello. La primera instalación de Zabbix correspondía a la versión **1.6.3**.

| Actualización | Motivo |
|---|---|
| **Versión 1.6.3 a versión 1.6.4** | Negación de servicio y vulnerabilidades a nivel de inyección de código SQL.<br><br>Para más información, ver **[15]**. |
| **Versión 1.6.4 a versión 1.8** | Mejoras en el frontend de Zabbix (soporta mayor número de elementos monitorizados sin merma en las prestaciones del sistema), soporte para versión 5.3 de PHP, creación de mapas y gráficos avanzados (ahora es posible arrastrar iconos a cualquier posición del mapa, mientras que en versiones anteriores era necesario especificar la ubicación de los mismos), mejoras en las reglas de descubrimiento (ofrece soporte para Ipv6), nuevas modalidades de monitorización sin agente basadas en Telnet y SSH.<br><br>Para más información, ver **[16]**. |
| **Versión 1.8 a versión 1.8.1** | Soluciona los problemas en la creación de gráficos y mapas (se producen errores en la base de datos a la hora de importar archivos de icono y de imagen).<br><br>Para más información, ver **[17]** y **[18]**. |
| **Versión 1.8.1 a 1.8.2** | Vulnerabilidades a nivel de inyección de código SQL.<br><br>Para más información, ver **[19]** y **[20]**. |

**Tabla 72. Listado de actualizaciones de Zabbix a lo largo del proyecto**

# 9

# Plan de contingencia

# 9. PLAN DE CONTINGENCIA

Los planes de contingencia se diseñan a modo de guía en la que reflejar las acciones que se deben realizar para recuperar la disponibilidad de un sistema una vez éste ha fallado. Son varias las causas por las que esto puede ocurrir, como por ejemplo, pérdida de los datos almacenados en el equipo, fallo del sistema operativo o bien fallos del hardware.

En el caso de la plataforma de monitorización implementada se han definido estrategias basadas en copias de seguridad que nos ayudarían a recuperar el sistema en caso de hipotéticas pérdidas de datos o fallos en el sistema operativo.

## 9.1. Backup de la base de datos MySQL

Prácticamente todo lo que Zabbix necesita se almacena en la base de datos, por lo que ésta es una de las partes del sistema de monitorización a la que más cuidado ha de prestarse. El método más utilizado para crear backups de la base de datos se basa en ***mysqldump***, utilidad nativa del propio MySQL.

Los **beneficios** que aporta esta estrategia son:

- Facilidad de configuración.
- El backup resultante puede funcionar bajo distintas versiones de MySQL.
- El propio backup se puede crear sin interferir en el funcionamiento del servidor MySQL.

Por el contrario, las **desventajas** que entraña esta solución son:

- Servidores con una alta carga pueden tardar mucho tiempo en crear el backup.
- Normalmente requiere espacio físico adicional en el que almacenar el fichero de backup creado.

```
$ mysqldump --add-drop-table --add-locks --extended-insert --single-transaction
--quick 'zabbix' | bzip2 > "fichero-de-backup".bz2
```

Este es el comando utilizado para crear backups de forma manual comprimidos en un fichero ".bz2[24]". Las opciones utilizadas son:

- **--add-drop-table**: añade sentencias de "drop" para que, en caso de restaurar la base de datos, no haya que borrar previamente las tablas ya existentes de forma manual.
- **--add-locks**: implicará operaciones de inserción más rápidas cuando se restaure la base de datos.
- **--extended-insert:** utiliza sentencias de inserción con varias filas y supondrá un menor tamaño del archivo de backup así como un proceso de restauración más rápido.
- **--single-transaction**: emplea una única transacción para todo el proceso de creación del backup, con lo cual éste puede tener un estado consistente sin bloquear ninguna tabla y sin provocar retrasos en el servidor Zabbix o en su frontend. Dado que Zabbix también utiliza transacciones para acceder a la base de datos, el backup creado debería estar siempre en un estado consistente.
- **--quick**: esta opción hace que *mysqldump* vaya tomando las filas de la base de datos una a una, en lugar de almacenarlas todas en un buffer intermedio, con lo cual se acelera el proceso de creación del backup para tablas grandes. Puesto que las tablas que contienen el historial de Zabbix suelen tener muchos registros almacenados, se recomienda activar esta opción a la hora de crear las copias de seguridad.

La política de backups de la base de datos Zabbix consiste en ejecutar el comando anterior de forma automatizada todos los días de la semana. Para ello, se incluye en el fichero **crontab**[25] de la máquina "zabbix-cctv" para que, todos los días a las 23 h se ejecute un script que invoque a dicho comando. El contenido de ese script se encuentra en ANEXO XII. Script de backup de la base de datos MySQL de Zabbix.

## 9.2. Backup de la máquina "zabbix-cctv"

La creación de copias de seguridad de la máquina en la que está alojada la plataforma de monitorización se basa en el software **Legato NetWorker Client [21]**, utilizado para crear los backups en los sistemas centrales de la UC3M.

[24] "bzip2" es un formato de compresión que alcanza mayor porcentaje de compresión que otros formatos de compresión más comunes como ".zip" o ".rar". En contrapartida, emplea más tiempo para su ejecución que esos otros formatos.
[25] Fichero que contiene los procesos que deben ejecutarse de forma regular según el administrador de procesos "cron" de los sistemas Linux.

Legato Networker es un sistema para creación de backups en red cuyo cliente se encuentra instalado en todos los equipos de la UC3M para los que están programadas copias de seguridad.

Para que el cliente pueda conectarse con el servidor de backup, es necesario que en el equipo en el que está instalado se esté ejecutando el demonio **nsrexecd**, cuya configuración se indica en un fichero habitualmente alojado en el directorio */etc/init.d/networker* del equipo cliente. El contenido de un fichero ejemplo de configuración "networker" puede verse en ANEXO V. Ejemplo de fichero de configuración networker.

# 10

# Conclusiones y líneas futuras

# 10. CONCLUSIONES Y LÍNEAS FUTURAS

Dentro de este capítulo se mostrarán las conclusiones a nivel tecnológico, el trabajo futuro a seguir en la línea de monitorización y la valoración personal que supone el haber llevado a cabo este proyecto.

## 10.1. Conclusiones tecnológicas

Nos centraremos aquí en los objetivos tecnológicos alcanzados con la construcción de la plataforma de monitorización. En el apartado de introducción del presente documento se hablaba de la necesidad de encontrar una manera con la que poder estar informados en todo momento del estado del sistema de videovigilancia en la universidad.

A día de hoy disponemos de una plataforma de monitorización que cubre las necesidades originales, disponible las 24 horas del día durante los 7 días de la semana y registrando un histórico de datos que nos permiten conocer no sólo el estado actual del sistema sino su evolución a lo largo del tiempo que lleva desplegada la plataforma. Pero lo más importante de todo es que ahora, frente a la época en la que la monitorización del sistema de videovigilancia dependía enteramente de la intervención humana, el control es muchísimo mayor y con muchísimo menos esfuerzo. Es obvio que el hecho de construir la plataforma de monitorización ha supuesto un coste en recursos, ya sean económicos o humanos, pero también lo es que la mejora global obtenida en el funcionamiento del sistema de CCTV de la universidad compensa esos costes. Es justo reconocer que el sistema de videovigilancia implementado actualmente en la universidad tiene sus limitaciones a nivel físico y a nivel de diseño y que la plataforma de monitorización, por sí sola, no puede solventarlas, pero no cabe duda de que, con ayuda de la plataforma, podemos mejorar el rendimiento del sistema vigente.

Por otra parte, Zabbix se ha revelado como un software de monitorización muy útil, incluso más flexible y sencillo de configurar que el resto de software similar que se evaluó en un principio. Con seguridad podemos decir que Zabbix es una de las mejores soluciones en la línea de monitorización para aquellos usuarios que tienen un primer contacto con este tipo de herramientas. El hecho de no tener que editar continuamente ficheros de configuración y disponer de una interfaz web desde la que personalizar multitud de parámetros de funcionamiento, unido todo ello a la existencia de foros de usuarios y manuales con los que se resuelven rápidamente las dudas, supone que, a día de hoy, se considere a Zabbix como una elección acertada.

Actualmente se monitorizan un total de **8973** items repartidos en **294** equipos o hosts en total. El número de disparadores que evalúan los valores de monitorización para lanzar las alertas correspondientes en caso de que sea necesario es de **1457**. Con el

tiempo, esas cifras crecerán, y su evolución se podrá observar en los gráficos con los cuenta Zabbix.

En resumen, el despliegue de la plataforma de monitorización basada en Zabbix supone un gran apoyo para el área de Seguridad del Servicio de Informática de la universidad y para el personal de seguridad física, ya sea para el día a día, ayudando a anticiparse a las incidencias antes de que se produzcan o a resolverlas una vez que han tenido lugar, o para futuras mejoras que se deseen implantar en el sistema de videovigilancia. Con los datos de monitorización es más sencillo tomar decisiones concernientes a un sistema tan importante como es el de la videovigilancia.

## 10.2. Líneas futuras

Enumeramos a continuación las ideas de trabajo futuro que desean seguirse en el proceso de monitorización:

- **Replicación de la base de datos de Zabbix**

  El servidor central de Zabbix "zabbix-cctv" cuenta con dos discos SCSI de 300 GB con los cuales se configura una unidad RAID 1 **[27]**. Así, al tener dos discos en espejo, garantizamos que en todo momento se realiza una copia de los datos que se están modificando. De esta manera, junto con el backup de la base de datos programado diariamente y las copias de seguridad en el servidor de backup de la UC3M, aseguramos la redundancia de los datos de monitorización. Sin embargo, existe otra solución no contemplada actualmente y es la replicación de la base de datos MySQL, para lo cual sería necesario utilizar un servidor adicional que hiciera las veces de servidor esclavo **[28]**. Si, por alguna razón, nuestro servidor maestro ("zabbix-cctv") fallara, el servidor esclavo podría pasar a sustituirlo de forma automática. Adoptando esta solución, nos aseguraríamos de que, a la hora de de restaurar el sistema en caso de fallo, no habría tiempos muertos ni paradas de ningún tipo.

  Otros de los aspectos en los que la replicación beneficiaría es el rendimiento del propio servidor MySQL. Como dijimos anteriormente, el motor de almacenamiento utilizado para la base de datos de Zabbix es MyISAM. Para este tipo concreto de motor, si el número de operaciones de escritura es proporcionalmente menor al número de operaciones de lectura, puede considerarse la replicación como una manera de optimizar el rendimiento, haciendo que las escrituras se realicen sobre el *maestro* y las lecturas sobre el servidor *esclavo*, reduciendo así la carga sobre el primero.

- **Integración de los equipos del control de accesos**

  El sistema de control de accesos está gobernado por una serie de equipos denominados como *CPU's* a las que es posible conectarse vía HTTP. Se estudiará de qué manera se pueden obtener parámetros de monitorización relevantes para conocer el estado de estos equipos.

- **Mejora del almacenamiento**

  Se pretende implantar un sistema de almacenamiento en los servidores distinto al implementado actualmente en forma de un RAID para cada servidor. La idea es implementar una solución centralizada para todos los servidores que también sería monitorizada con Zabbix.

- **Envío de alertas vía SMS**

  Las acciones creadas hasta ahora en Zabbix envían notificaciones vía e-mail, pero en el futuro se evaluará enviar dichas notificaciones también a través de mensajes cortos SMS aprovechando el soporte que Zabbix ofrece a este mecanismo de alertas.

- **Monitorización con 'orabbix' de la instancia Oracle de la futura base de datos del control de accesos**

  Los datos del control de accesos serán migrados en breve a una instancia Oracle que podrá ser monitorizada con el plugin '*orabbix*' [**36**] disponible para Zabbix.

- **Creación de un cuadro de mando orientado a los usuarios menos frecuentes de la plataforma**

  Ya hemos mencionado en secciones anteriores que Zabbix permite la creación de gráficos personalizados con cualquier conjunto de iconos y fondos que el usuario imagine. Actualmente, a modo de vistazo general, se cuenta con el siguiente gráfico para indicar el estado de los servidores de grabación:

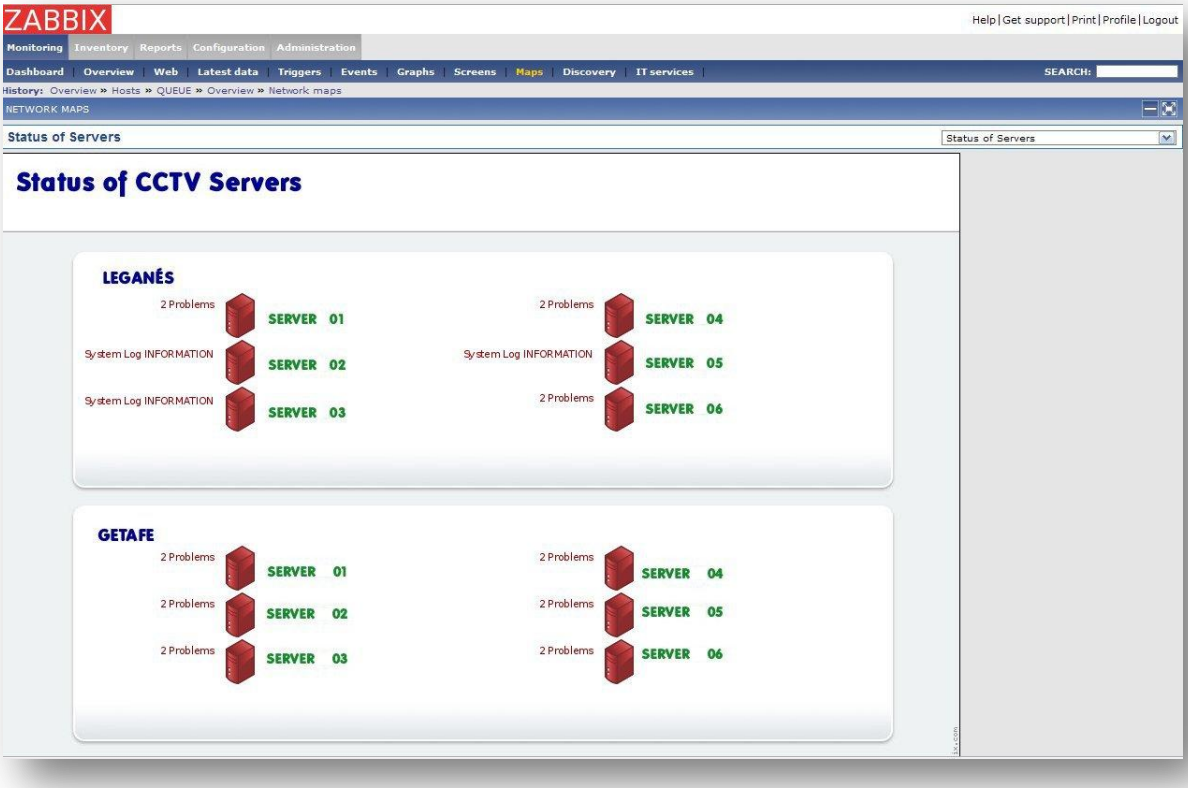



**Figura 36. Gráfico del estado general de los servidores de grabación**

En este gráfico, los servidores adoptan un color azulado cuando no se registra en ellos ninguna alarma y el color rojo de la figura cuando, por el contrario, se produce algún tipo de evento sobre los mismos. En el caso de las cámaras existen planos por cada planta de cada edificio con un icono situado sobre el plano con la ubicación real de cada cámara.

Se está considerando diseñar un nuevo gráfico más intuitivo para usuarios que no estén tan familiarizados con la plataforma para que puedan observar el funcionamiento de los componentes principales (servidores y cámaras) del sistema de CCTV de la universidad.

- **Monitorización del servidor MySQL**

  Actualmente se está trabajando en la monitorización del servidor MySQL instalado en la máquina "zabbix-cctv". La solución adoptada se encuentra en una sección concreta del sitio oficial de Zabbix **[26]** y se basa en la ejecución de un script escrito en PHP que recopila parámetros de configuración del servidor MySQL. La idea buscada es obtener toda la información posible para seguir ajustando el rendimiento de dicho servidor.

- **Interpretación de los errores AEN recogidos por la controladora 3ware**

  La controladora 3ware gestiona el sistema RAID de almacenamiento en cada servidor de grabación y registra en los eventos del sistema operativo Windows multitud de mensajes AEN [37] en los que se indican los errores producidos junto con un código para identificarlos. Actualmente esos mensajes AEN se notifican como un evento más de los logs de Windows. En el futuro se desea interpretar esos mensajes AEN para monitorizarlos de forma independiente a los eventos registrados por los sistemas operativos Windows.

- **Creación de una guía de usuario de la plataforma de monitorización implementada**

  Se redactará una guía de usuario que describa cómo operar con la plataforma de monitorización implementada en Zabbix.

## 10.3. Valoración personal

Puedo decir con toda seguridad que el hecho de haber tomado parte en este proyecto me ha aportado beneficios no sólo profesionales sino también personales. Comenzar un proyecto desde cero y ser, además de responsable del mismo, testigo directo de cómo va evolucionando, me ha supuesto una motivación adicional para seguir adelante en aquellos momentos en que surgía algún tipo de problema. De todas formas, no hay nada como ser consciente de que aquello en lo que estás trabajando realmente sirve para algo y ello anima a seguir invirtiendo tu tiempo.

Es gratificante saber que, con un proyecto de estas características, se están solucionando, si bien no todos, numerosos problemas que afectaban al rendimiento de un sistema tan importante como es el de la videovigilancia en la universidad. Soy plenamente consciente de que hay aspectos que se pueden mejorar y de que surgirán nuevas necesidades que probablemente impliquen obligadas modificaciones en el proyecto, pero, como decía, el saber que es algo útil te aporta esas ganas de seguir trabajando en una línea de mejora.

A nivel profesional son muchos los conocimientos adquiridos a lo largo de todo este tiempo. El más importante de todos ellos es la experiencia adquirida en el manejo ya no sólo de un sistema de monitorización particular como es Zabbix, sino también otros como Nagios y Pandora FMS. Pero no acaba ahí, ya que el trabajo desarrollado me ha permitido ganar experiencia en aspectos como la administración de sistemas UNIX y Windows, configuración y mantenimiento de bases de datos MySQL, configuración y administración del software y las cámaras del fabricante Sony dentro del mundo de la videovigilancia, protocolo SNMP, sistemas de almacenamiento, políticas sobre copias de seguridad, creación de scripts, por citar algunos ejemplos.

Por último, y no por ello menos importante, el haber desarrollado este proyecto me ha ayudado a adquirir el hábito de documentarme para así afrontar con garantías la resolución de cualquier problema.

# 11

# Bibliografía

# 11. BIBLIOGRAFÍA

El material consultado a lo largo del proyecto es el siguiente:

## 11.1. Libros

Rihards Olups: ***'Zabbix 1.6 Network Monitoring: Monitor your network's hardware, servers and web performance effectively and efficiently'***. Publicado por Packt Publishing Ltd (www.packtpub.com). Octubre 2009. ISBN 978-1-847197-68-9.

Jeremy D. Zawodny y Derek J. Balling: ***'MySQL Avanzado: Optimización, copias de seguridad, replicación y equilibrado de carga'***. Publicado por O'Reilly, 2004. ISBN 84-415-1759-2.

Wojciech Kocjan: ***'Learning Nagios 3.0: A detailed tutorial to setting up, configuring and managing this easy and effective system monitoring software'***. Publicado por Packt Publishing Ltd (www.packtpub.com). Octubre 2008. ISBN 978-1-84719-518-0.

William von Hagen y Brian K. Jones: ***'Linux Server Hacks, Volume Two: Tips & Tools for Connecting, Monitoring and Troubleshooting'***. Publicado por O'Reilly. Diciembre 2005. ISBN 978-0596100827.

James Turnbull, Peter Lieverdink y Dennis Matotek: ***'Pro Linux System Administration: The complete guide to Linux administration'***. Publicado por Apress. Junio 2009. ISBN 978-1430219125.

## 11.2. Páginas/Documentos electrónicos en la red

***Documentación oficial de Zabbix.***
http://www.zabbix.com/documentation/
[Último acceso en Agosto de 2010].

***Documentación oficial de Pandora FMS.***
http://pandorafms.org/index.php?sec=project&sec2=documentation&lng=es
[Último acceso en Agosto de 2010].

***Documentación oficial de Nagios.***
http://www.nagios.org/documentation
[Último acceso en Agosto de 2010].

***RealShot Manager User and Administrator Guide* (versión en inglés).**
*http://pro.sony.com/bbsccms/assets/files/cat/camsec/downloads/RealShot_Manager_User_And_Administrator_Guide_EN.pdf*
[Último acceso en Junio de 2010].

***Estándar de Ingeniería de Software de la European Space Agency (ESA).***
Sergio Ochoa, M.Cecilia Bastarrica.
*http://www.face.ubiobio.cl/~cgutierr/clase04-2.pdf*
[Accedido en Mayo de 2010].

***Agencia Espacial Europea (ESA).***
*http://www.esa.int/esaCP/Spain.html*
[Accedido en Mayo de 2010].

***ZABBIX Forums***
*http://www.zabbix.com/forum/*

## 11.3. Normas

***Métrica. Versión 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información.***
Ministerio de Administracions Públicas.
*http://www.csae.map.es/csi/metrica3/*
[Último acceso en Agosto de 2010].

***Electrical Industries Association/Telecommunications Industries Association***
*EIA/TIA 568A (1994).*
[Último acceso en Mayo de 2010].

***International Standarization Organization/Inter. Electrotechnical Comission"***
*ISO/IEC 11801 (1994).*
*Adoptado por CENELEC (Comité Europeo de Normalización Eléctrica) EN 50173 con ligeras variantes.*
[Último acceso en Mayo de 2010].

# 12

# Definiciones y acrónimos

# 12. DEFINICIONES Y ACRÓNIMOS

Se incluye aquí un listado con los términos y abreviaturas utilizados a lo largo del documento y cuya explicación se ha estimado oportuna.

## 12.1. Definiciones

- **Agente**
  Parte de un software que actúa para un usuario o un programa y que tiene capacidad para decidir cómo hacerlo para alcanzar unos objetivos.
- **AIX**
  Sistema operativo basado en Unix propietario de IBM.
- **Backend**
  Dentro de una aplicación software, es la parte que procesa los datos de entrada que llegan desde otra capa conocida como *frontend*.
- **Bz2**
  Extensión de los archivos comprimidos con el comando *bzip2*, aplicación desarrollada para comprimir y descomprimir archivos.
- **Capa de aplicación**
  Se trata del séptimo nivel dentro del modelo OSI y ofrece a las aplicaciones la posibilidad de acceder a los servicios de las restantes capas del modelo. Define los protocolos que utilizan las aplicaciones para intercambio de información.
- **Disco duro SCSI**
  Disco duro provisto de interfaz SCSI (*Small Computer System Interface*). Esta interfaz está preparada para grandes capacidades de almacenamiento y velocidades de rotación.
- **Fast Ethernet**
  Estándar de transmisión de redes LAN que proporciona una velocidad de transmisión de 100 megabits por segundo (100BASE-T).
- **Firewall**
  Componente de un sistema informático a través del cual se restringe el acceso a éste, bloqueando a los equipos no autorizados.
- **Forwarding**
  A nivel de redes informáticas, mecanismo que permite redireccionar o encaminar el tráfico.

- **FreeBSD**
  Sistema operativo derivado de BSD, la versión de Unix desarrollada en la Universidad de Berkeley (California).
- **Frontend**
  Dentro de una aplicación software, el frontend es la capa que interactúa con los usuarios.
- **Gateway**
  Dispositivo a través del cual se interconectan redes informáticas.
- **Gigabit Ethernet**
  Ampliación del estándar Ethernet para conseguir una capacidad de transmisión de 1 gigabit por segundo.
- **H.264**
  Norma que define un códec de vídeo de alta compresión nacida con la intención de crear un estándar capaz de proporcionar calidad de imagen con tasas binarias inferiores a estándares existentes como MPEG-4.
- **Host**
  Término utilizado para hacer referencia a un equipo informático conectado a una red cualquiera.
- **Hot spare**
  En un sistema RAID, los discos configurados como *hot spare* actúan como discos de reserva para entrar en funcionamiento en el momento que uno o más discos del RAID fallan.
- **HP-UX**
  Sistema operativo basado en UNIX desarrollado y mantenido por el fabricante *Hewlett-Packard*.
- **InnoDB**
  Motor de almacenamiento para bases de datos MySQL que opera a nivel de transacciones.
- **Iptables**
  Herramienta de *Netfilter* **[41]** que permite configurar un firewall en base a unas políticas de filtrado de tráfico.
- **JPEG**
  Formato de compresión de imágenes. Permite graduar el nivel de compresión (mayor o menor calidad) basándose para ello en reducir información promediándola en las zonas de degradado.
- **Log**
  En el plano de la informática, un log es un registro de eventos, información y datos.

- **Make**
  Herramienta utilizada en los sistemas UNIX para la compilación de programas.
- **Mantenimiento correctivo**
  Cambios precisos para corregir errores en un producto una vez se han producido.
- **Mantenimiento evolutivo**
  Modificaciones necesarias para que un producto cubra cambios en las necesidades.
- **Mantenimiento preventivo**
  Operaciones llevadas a cabo con objeto de evitar que se produzcan fallos en un producto.
- **MPEG**
  Estándar de compresión de archivos de audio y vídeo.
- **MPEG4**
  Estándar de compresión de audio y vídeo diseñado especialmente para codificación con bitrate inferior a 1.5 Mbit por segundo.
- **MyISAM**
  Motor de almacenamiento utilizado por defecto en bases de datos MySQL. Cada tabla MyISAM se almacena en disco en 3 ficheros (fichero de datos, fichero de definición de tabla, fichero de índices).
- **OpenBSD**
  Sistema operativo libre multi-plataforma basado en BSD y enfocado en la seguridad y la criptografía.
- **Open Source**
  Término utilizado para hacer referencia al software que es distribuido y desarrollado libremente.
- **OS X**
  Sistema operativo desarrollado por el fabricante *Apple* basado en UNIX.
- **PostgreSQL**
  Sistema de gestión de bases de datos relacionales publicado bajo la licencia BSD. Destaca por tener unas prestaciones y funcionalidades equivalentes a otros gestores de bases de datos comerciales.
- **SCO Open Server**
  Sistema operativo creado por el grupo *SCO Group* basado en UNIX.
- **Servidor**
  Dentro de una red informática, el servidor es el equipo que proporciona servicios al resto de equipos (clientes) conectados a la red.

- **Sistema de monitorización**
  En informática, un sistema de monitorización es una herramienta que permite analizar el estado de diferentes parámetros de una serie de equipos conectados a una red.
- **Sistema RAID**
  Sistema de almacenamiento formado por un conjunto de discos duros y que se implementa con objeto de conseguir redundancia y tolerancia a fallos en los datos almacenados.
- **SNMP trap**
  Mensaje iniciado por un elemento de red (un agente) y enviado al sistema administrador de la red (usuario SNMP). También es común definirlo como un evento asíncrono generado por un agente SNMP.
- **snmpwalk**
  Función que se utilizar para leer valores de un agente SNMP.
- **Software libre**
  Término utilizado para denominar el software que puede ser usado, copiado, estudiado, cambiado y redistribuido libremente por parte de sus usuarios.
- **Solaris**
  Sistema operativo desarrollado inicialmente por *Sun Microsystems* y actualmente por *Oracle Corporation* catalogado oficialmente como una versión más de UNIX.
- **Telnet**
  Protocolo que proporciona en redes LAN comunicaciones bidireccionales a través de un terminal virtual de red (NVT). Permite que un usuario pueda conectarse con cualquier otro host independientemente del sistema operativo.
- **Terminal virtual de red**
  Dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada equipo mapea las características de su propia terminal sobre las de su correspondiente terminal virtual de red.
- **Topología de red**
  Disposición física que refleja la manera en que se encuentran conectados los equipos de una red.
- **Tuning**
  En el contexto de bases de datos, este término hace referencia a un conjunto de operaciones encaminadas a optimizar el rendimiento.

- **Trends**
  En los sistemas de monitorización, los trends son los datos con los que se construyen las estadísticas de los equipos monitorizados.
- **Trunk**
  En el contexto de redes informáticas, el término *trunk* hace referencia a una conexión de red que transporta múltiples VLANs.

## 12.2. Acrónimos

- **AIX**
  *Advanced Interactive eXecutive.*
- **BSD**
  *Berkeley Software Distribution.*
- **CCTV**
  *Circuito Cerrado de Televisión.*
- **CGI**
  *Common Gateway Interface.*
- **CPU**
  *Central Processing Unit.*
- **DHCP**
  *Dynamic Host Configuration Protocol.*
- **FMS**
  *Flexible Monitoring System.*
- **GCC**
  *GNU C Compiler.*
- **GNU**
  *GNU's Not Unix* (acrónimo recursivo).
- **GPL**
  *General Public License* (Licencia Pública General de GNU).
- **GUI**
  *Graphical User Interface.*
- **HTTP**
  *HyperText Transfer Protocol.*

- **ICMP**

  *Internet Control Message Protocol.*

- **JPEG**

  *Joint Photographic Experts Group* (Grupo conjunto de expertos en fotografía).

- **LAN**

  *Local Area Network.*

- **MIB**

  *Management Information Base.*

- **MJPEG**

  *Motion JPEG.*

- **NTP**

  *Network Time Protocol.*

- **NVT**

  *Network Virtual Terminal.*

- **PHP**

  *PHP Hypertext Pre-processor* (acrónimo recursivo).

- **RAID**

  *Redundant Array of Independent Disks.*

- **RTT**

  *Round-Trip delay Time.*

- **SLA**

  *Service Level Agreement.*

- **SMI**

  *Structure of Management Information.*

- **SMS**

  *Short Message Service.*

- **SMTP**

  *Simple Mail Transfer Protocol.*

- **SNMP**

  *Simple Network Management Protocol.*

- **SQL**

  *Structured Query Language.*

- **SSL**

  *Secure Sockets Layer.*

- **TCP**

  *Transmission Control Protocol.*

- **UDP**

  *User Datagram Protocol.*

- **VLAN**

  *Virtual LAN.*

- **WMI**

  *Windows Management Instrumentation.*

- **XML**

  *Extensible Markup Language.*

# 13

# Referencias

# 13. REFERENCIAS

A lo largo del documento se han ido enumerando una serie de referencias a determinadas fuentes de información seguidas para la realización del proyecto. El formato escogido para esas referencias, como se ha podido comprobar tras la lectura del presente documento, es "[X]", donde X es una cifra que indica el orden de la referencia.

En la siguiente tabla mostraremos el número de referencia, así como el contenido y la fuente correspondientes:

| REFERENCIA | CONTENIDO | FUENTE |
|---|---|---|
| [1] | **Web oficial de ZABBIX** | *http://www.zabbix.com/* |
| [2] | **Protocolo UDP** | *http://www.networksorcery.com/enp/protocol/udp.htm* |
| [3] | **Librerías GD** | *http://www.boutell.com/gd/* |
| [4] | **Licencias GPL** | *http://www.gnu.org/licenses/licenses.es.html* |
| [5] | **Web oficial de Nagios** | *http://www.nagios.org/* |
| [6] | **Software Sony RealShot Manager™** | *http://www.sony.es/biz/view/ShowProduct.action?product=RealShot+Manager+V4&pageType=Overview&category=NVMMonSoftware* |
| [7] | **Software Supero Doctor III** | *http://www.supermicro.com/products/accessories/software/superodoctoriii.cfm* |
| [8] | **Web oficial de PostgreSQL** | *http://www.postgresql.org/* |
| [9] | **Montura *Ball-Joint* patentada por Sony** | *http://www.sony.es/biz/view/ShowContent.action?site=biz_es_ES&category=NVMMinidomes&contentId=1181893266026&sectiontype=Product&preserveContext=true* |
| [10] | **Métrica V3** | *http://www.csi.map.es/csi/metrica3/index.html* |
| [11] | **Plugin *nagvis* para Nagios** | *http://www.nagvis.org/* |
| [12] | **Web de *checkinstall*** | *http://www.asic-linux.com.mx/~izto/checkinstall/* |
| [13] | **Motor de almacenamiento MyISAM** | *http://dev.mysql.com/doc/refman/5.0/es/myisam-storage-engine.html* |

| | | |
|---|---|---|
| [14] | Utilidad *mytop* para MySQL | http://jeremy.zawodny.com/mysql/mytop/mytop.html |
| [15] | ZABBIX Denial Of Service and SQL Injection Vulnerabilities | http://www.securityfocus.com/bid/37309 |
| [16] | Release notes versión 1.8 de ZABBIX | http://www.zabbix.com/rn1.8.php |
| [17] | Bug sobre creación de imágenes en la versión 1.8 de ZABBIX | http://www.zabbix.com/forum/showthread.php?t=14795 |
| [18] | Bug sobre creación de imágenes en la versión 1.8 de ZABBIX | https://support.zabbix.com/browse/ZBX-1620?page=com.atlassian.jira.plugin.system.issuetabpanels%3Achangehistory-tabpanel |
| [19] | Vulnerabilidad de código SQL en la versión 1.8.1 de ZABBIX | http://archives.neohapsis.com/archives/fulldisclosure/2010-04/0001.html |
| [20] | Vulnerabilidad de código SQL en la versión 1.8.1 de ZABBIX | https://support.zabbix.com/browse/ZBX-2257 |
| [21] | Cliente Legato Networker | http://www.emc.com/products/detail/software/networker.htm |
| [22] | Web oficial de las cámaras de videovigilancia Sony | http://www.sony.es/biz/product/nvmcameras |
| [23] | Modelo de cámara de videovigilancia SNC-CS50P | http://www.sony.es/biz/product/nvmfixedcameras/snc-cs50p/overview |
| [24] | Modelo de cámara de videovigilancia SNC-DF80P | http://www.sony.es/biz/product/nvmminidomes/snc-df80p/overview |
| [25] | Modelo de cámara de videovigilancia SNC-RX550P | http://www.sony.es/biz/product/nvmptzcameras/snc-rx550p-bc/overview |
| [26] | Heavy MySQL Monitoring Solution | http://www.zabbix.com/wiki/howto/monitor/db/mysql/extensive_mysql_monitoring_including_replication |
| [27] | Niveles de RAID | http://www.smdata.com/NivelesRAID.htm |
| [28] | MySQL Replication | http://dev.mysql.com/doc/refman/4.1/en/replication.html |
| [29] | Web oficial de Pandora FMS | http://pandorafms.org/ |

| | | |
|---|---|---|
| [30] | Manual de Pandora FMS | http://sunet.dl.sourceforge.net/project/pandora/Pandora%20FMS%203.0/Final%20version%20%28Stable%29/PandoraFMS_Manual_ES_RC1.pdf |
| [31] | Definición de umbrales de alerta en Nagios | http://nagiosplug.sourceforge.net/developer-guidelines.html#THRESHOLDFORMAT |
| [32] | User Parameters en Zabbix | http://www.zabbix.com/documentation/1.8/manual/config/user_parameters |
| [33] | SNMP Manpage | http://net-snmp.sourceforge.net/docs/man/snmpwalk.html |
| [34] | check_snmp (plugin Nagios) | http://nagiosplugins.org/man/check_snmp |
| [35] | State Types en Nagios | http://nagios.sourceforge.net/docs/2_0/statetypes.html |
| [36] | Plugin 'orabbix' para Zabbix | http://freshmeat.net/projects/orabbix |
| [37] | AEN Messages (controladora RAID 3ware) | https://www.3ware.com/3warekb/attachments/Pages%20from%209000%20series%20UsrGuide.pdf |
| [38] | Windows Management Instrumentation (WMI) | http://msdn.microsoft.com/en-us/library/aa394582%28VS.85%29.aspx |
| [39] | Zabbix Forums (foro oficial de Zabbix) | http://www.zabbix.com/forum/ |
| [40] | Blog de Zabbix en español | http://zabbix-es.blogspot.com/ |
| [41] | Site oficial de Netfilter/iptables | http://www.netfilter.org/ |
| [42] | Site oficial de Ubuntu Linux | http://www.ubuntu.com/ |
| [43] | Remote Commands (comandos remotos) en Zabbix | http://www.zabbix.com/documentation/1.8/manual/tutorials/remote_actions |

**Tabla 73. Referencias utilizadas**

# 14

# Anexos

# 14. ANEXOS

Se incluyen aquí los anexos que aportan información adicional al contenido del proyecto.

## 14.1. ANEXO I. Configuración del frontend Web de Zabbix

La interfaz Web de la que está provisto Zabbix se debe configurar desde un explorador web escribiendo *http://<nombre_del_servidor_o_direccion_ip/zabbix* en la barra de direcciones.

Se nos mostrará en primer lugar una pantalla de bienvenida. Pulsamos sobre el botón *Next*.

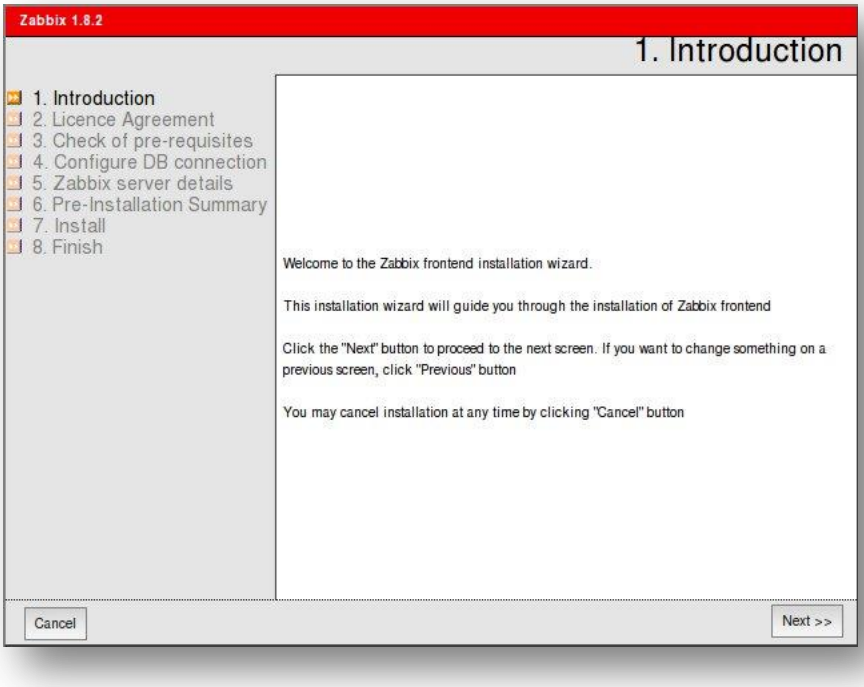



**Figura 37. Inicio de la instalación del frontend de Zabbix**

A continuación aparecerá una segunda pantalla con los acuerdos de licencia y pulsaremos nuevamente sobre *Next* para ir al siguiente paso.

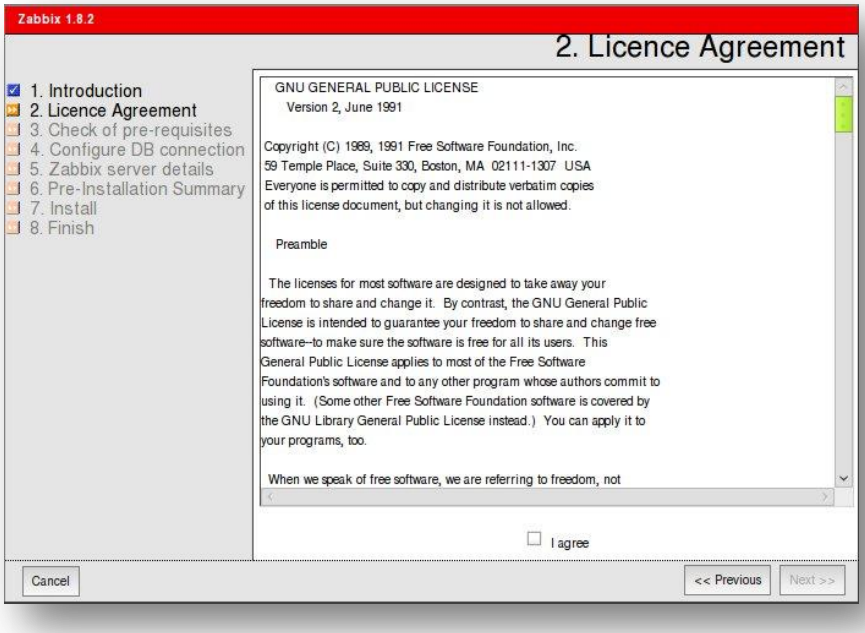



**Figura 38. Acuerdo de licencia del frontend de Zabbix**

Llegamos al tercer punto, antes del cual deberemos haber configurado correctamente todos los parámetros PHP. Las entradas ***PHP versión, PHP Databases support, PHP BC math support, GD Version*** y ***Image formats*** se refieren a la instalación del propio PHP, por lo que, si presentaran problemas, se resolverían instalando/reinstalando los paquetes apropiados (*php5-bcmath, php5-gd, php5-mysql*, etc) o recompilando PHP con las opciones correspondientes.

***PHP Memory limit, PHP post max size, PHP max execution time*** y ***PHP timezone*** son parámetros de configuración que podemos localizer en el archivo *php.ini*. Este archivo normalmente está ubicado en */etc/php5*.

En caso de tener problemas para localizar ese archivo de configuración, crearemos un fichero de prueba que incluya este contenido:

**<?php phpinfo() ¿>**

Así, bastará acceder a este fichero desde el navegador Web para buscar la entrada ***Configuration File (php.ini) Path***, que es donde deberíamos buscar el archivo *php.ini*.

Si hasta ahora no ha habido ningún problema en cuanto a PHP, la pantalla que nos encontraremos será la siguiente:

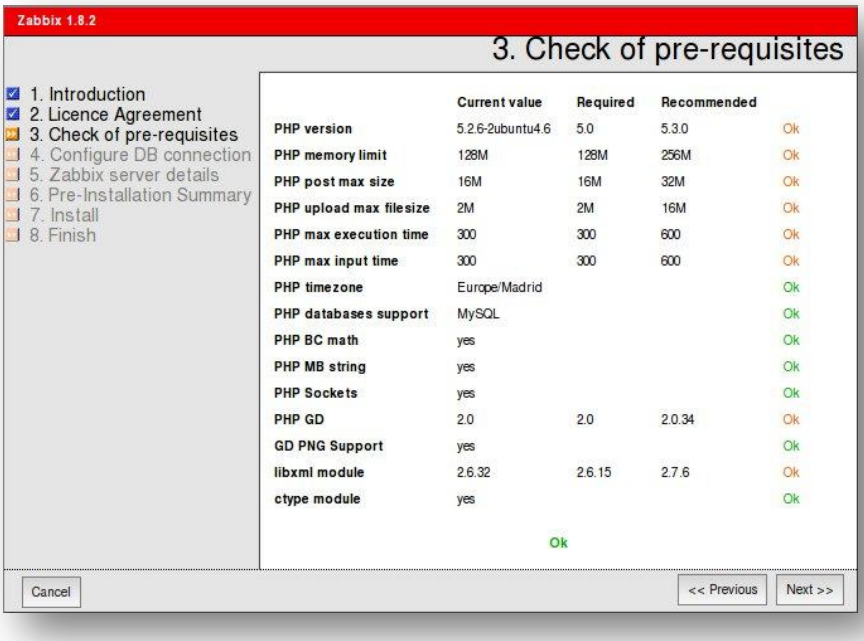



**Figura 39. Comprobación de pre-requisitos del frontend de Zabbix**

Seguiremos con la configuración de la conexión a la base de datos, introduciendo para ello el nombre, usuario y contraseña correspondientes. Pulsamos sobre *Test* para probar la conexión y, en caso de que no haya fallo, continuaremos la configuración del frontend pulsando el botón *Next*.

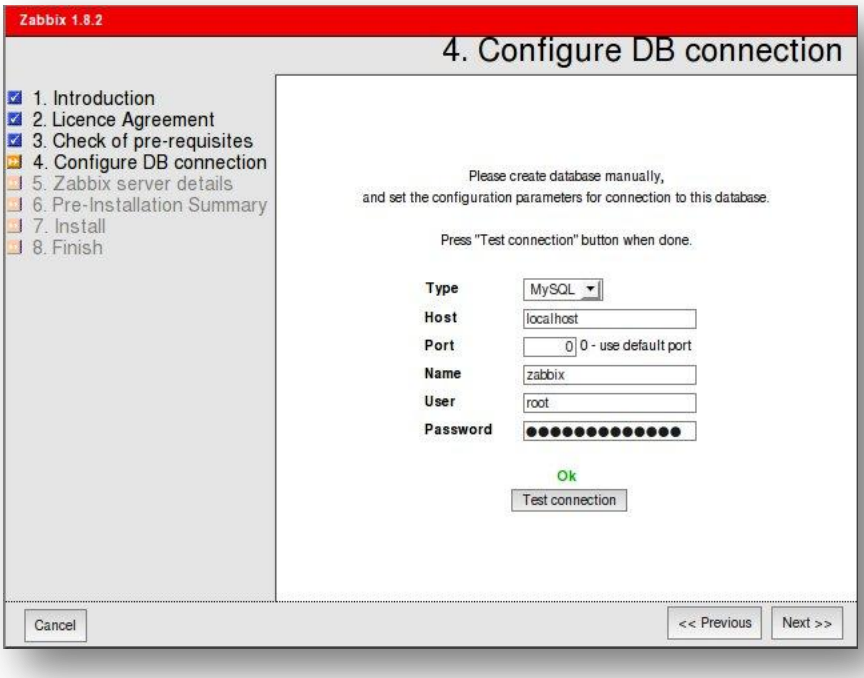



**Figura 40. Configuración de la conexión a la base de datos Zabbix**

Mantenemos la configuración predeterminada para el servidor Zabbix y pulsamos sobre *Next*.

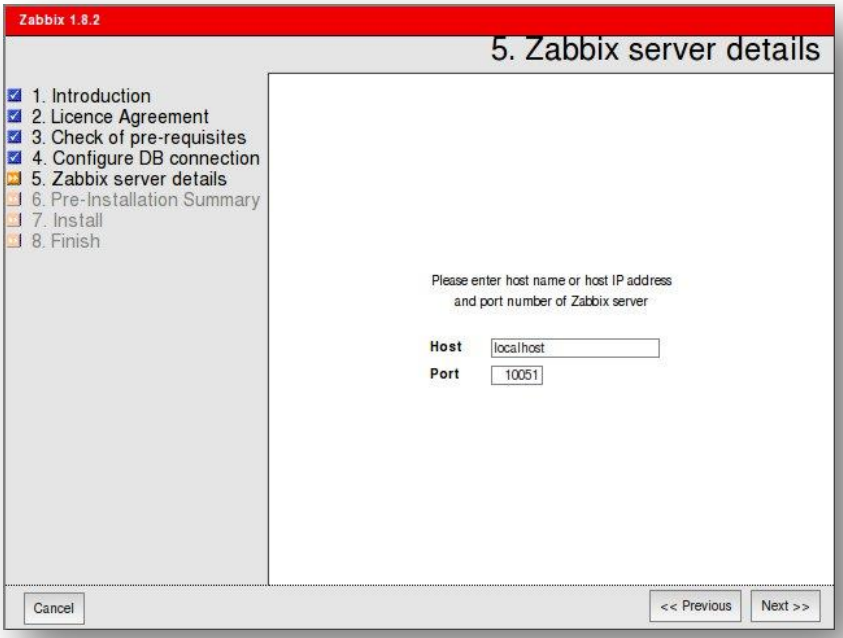



**Figura 41. Detalles de la configuración del servidor Zabbix**

La siguiente pantalla que veremos es un resumen de las configuraciones establecidas en las dos pantallas anteriores. Pulsamos sobre *Next*.

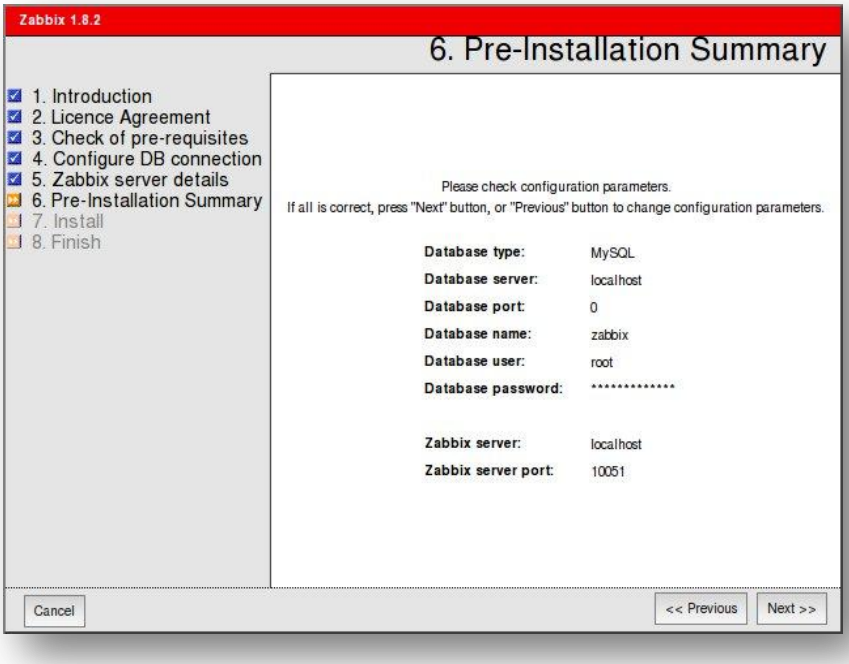



**Figura 42. Resumen de la instalación del frontend de Zabbix**

En este momento, la instalación nos indicará un error, avisando que no es posible modificar los datos en el fichero de configuración. La razón es que el fichero de configuración no tiene los permisos correspondientes, y lo solucionaremos descargando este fichero pulsando sobre el botón *Save Configuration file*. Una vez lo hayamos descargado, lo ubicaremos en el directorio */var/www/htdocs/zabbix/conf* y pulsaremos sobre el botón *Retry*. En este momento ya podremos continuar pulsando en *Next*.


**Figura 43. Descarga del fichero de configuración del servidor Zabbix**

Finalmente, aparecerá la última pantalla en la que se nos indicará que ha concluido la instalación del frontend de Zabbix.


**Figura 44. Fin de la instalación del frontend de Zabbix**

# 14.2. ANEXO II. Claves de monitorización en Zabbix

## Claves soportadas por el sistema operativo

| Parámetro | | Windows | Linux 2.4 | Linux 2.6 |
|---|---|---|---|---|
| agent.ping | | X | X | X |
| agent.version | | X | X | X |
| kernel.maxfiles | | - | X | X |
| kernel.maxproc | | - | - | X |
| log[file,<regexp>,<encoding>,<maxlines>,<mode>] | | X | X | X |
| logrt[file_format,<regexp>,<encoding>,<maxlines>,<mode>] | | X | X | X |
| eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>] | | X | - | - |
| net.if.collisions[if] | | - | X | X |
| net.if.in[if,<mode>] | | X | X | X |
| mode | bytes | X | X | X |
| | packets | X | X | X |
| | errors | X | X | X |
| | dropped | X | X | X |
| net.if.list | | X | - | - |
| net.if.out[if,<mode>] | | X | X | X |
| mode | bytes | X | X | X |
| | packets | X | X | X |
| | errors | X | X | X |
| | dropped | X | X | X |
| net.if.total[if,<mode>] | | X | X | X |
| mode | bytes | X | X | X |
| | packets | X | X | X |
| | errors | X | X | X |
| | dropped | X | X | X |
| net.tcp.dns[<ip>,zone] | | - | X | X |
| net.tcp.dns.query[<ip>,zone,<type>] | | - | X | X |
| net.tcp.listen[port] | | X | - | - |
| net.tcp.port[<ip>,port] | | X | X | X |
| net.tcp.service.perf[service,<ip>,<port>] | | - | X | X |
| net.tcp.service[service,<ip>,<port>] | | - | X | X |

| | | | | |
|---|---|---|---|---|
| **proc.mem[<name>,<user>,<mode>,<cmdline>]** | | - | X | X |
| **mode** | **sum** | - | X | X |
| | **avg** | - | X | X |
| | **max** | - | X | X |
| | **min** | - | X | X |
| **proc.num[<name>,<user>,<state>,<cmdline>]** | | - | X | X |
| **state** | **all** | - | X | X |
| | **sleep** | - | X | X |
| | **zomb** | - | X | X |
| | **run** | - | X | X |
| **sensor[<temp>]** | | - | X | - |
| **temp** | **temp1** | - | X | - |
| | **temp2** | - | X | - |
| | **temp3** | - | X | - |
| **system.boottime** | | - | X | X |
| **system.cpu.intr** | | - | X | X |
| **system.cpu.load[<cpu>,<mode>]** | | X | X | X |
| **mode** | **avg1** | - | X | X |
| | **avg5** | - | X | X |
| | **avg15** | - | X | X |
| **system.cpu.num[<type>]** | | X | X | X |
| **type** | **online** | - | X | X |
| | **max** | - | X | X |
| **system.cpu.switches** | | - | X | X |
| **system.cpu.util[<cpu>,<type>,<mode>]** | | X | X | X |
| **type** | **user** | - | X | X |
| | **nice** | - | X | X |
| | **idle** | - | X | X |
| | **system** | - | X | X |
| | **kernel** | - | - | - |
| | **iowait** | - | - | X |
| | **interrupt** | - | - | X |
| | **softirq** | - | - | X |
| | **steal** | - | - | X |
| **mode** | **avg1** | - | X | X |
| | **avg5** | - | X | X |
| | **avg15** | - | X | X |

| | | | | |
|---|---|---|---|---|
| **system.run[command,<mode>]** | | X | X | X |
| **mode** | **wait** | X | X | X |
| | **nowait** | X | X | X |
| **system.hostname** | | X | X | X |
| **system.localtime** | | X | X | X |
| **type** | **utc** | X | X | X |
| | **local** | X | X | X |
| **system.swap.in[<swap>,<type>]** | | - | X | X |
| **type** | **count** | - | X | X |
| | **sectors** | - | X | X |
| | **pages** | - | X | X |
| **system.swap.out[<swap>,<type>]** | | - | X | X |
| **type** | **count** | - | X | X |
| | **sectors** | - | X | X |
| | **pages** | - | X | X |
| **system.swap.size[<swap>,<type>]** | | X | X | X |
| **mode** | **free** | - | X | X |
| | **total** | - | X | X |
| | **used** | - | X | X |
| | **pfree** | - | X | X |
| | **pused** | - | X | X |
| **system.uname** | | X | X | X |
| **system.uptime** | | X | X | X |
| **system.users.num** | | - | X | X |
| **vfs.dev.read[device,<type>,<mode>]** | | - | X | X |
| **type** | **sectors** | - | X | X |
| | **operations** | - | X | X |
| | **bytes** | - | - | - |
| | **sps** | - | X | X |
| | **ops** | - | X | X |
| | **bps** | - | - | - |
| **mode** | **avg1** | - | X | X |
| | **avg5** | - | X | X |
| | **avg15** | - | X | X |

| | | | | |
|---|---|---|---|---|
| vfs.dev.write[device,<type>,<mode>] | | - | X | X |
| type | sectors | - | X | X |
| | operations | - | X | X |
| | bytes | - | - | - |
| | sps | - | X | X |
| | ops | - | X | X |
| | bps | - | - | - |
| mode | avg1 | - | X | X |
| | avg5 | - | X | X |
| | avg15 | - | X | X |
| vfs.file.cksum[file] | | X | X | X |
| vfs.file.exists[file] | | X | X | X |
| vfs.file.md5sum[file] | | X | X | X |
| vfs.file.regexp[file,regexp,<encoding>] | | X | X | X |
| vfs.file.regmatch[file,regexp,<encoding>] | | X | X | X |
| vfs.file.size[file] | | X | X | X |
| vfs.file.time[file,<mode>] | | X | X | X |
| mode | modify | X | X | X |
| | access | X | X | X |
| | change | X | X | X |
| vfs.fs.inode[fs,<mode>] | | - | X | X |
| mode | total | - | X | X |
| | free | - | X | X |
| | used | - | X | X |
| | pfree | - | X | X |
| | pused | - | X | X |
| vfs.fs.size[fs,<mode>] | | X | X | X |
| mode | total | X | X | X |
| | free | X | X | X |
| | used | X | X | X |
| | pfree | X | X | X |
| | pused | X | X | X |

| | | | | |
|---|---|---|---|---|
| **vm.memory.size[<mode>]** | | X | X | X |
| **mode** | **total** | - | X | X |
| | **free** | - | X | X |
| | **shared** | - | X | X |
| | **buffers** | - | X | X |
| | **cached** | - | X | X |
| | **pfree** | - | X | X |
| | **available** | - | X | X |

**Tabla 74. Claves de monitorización soportadas por el sistema operativo**

## Claves soportadas por el agente Zabbix

| Key | Description | Parameters |
|---|---|---|
| **agent.ping** | Check the agent availability. | - |
| **agent.version** | Version of Zabbix Agent. | - |
| **kernel.maxfiles** | Maximum number of opened files supported by OS. | |
| **kernel.maxproc** | Maximum number of processes supported by OS. | |
| **log[file,<regexp>,<encoding>,<maxlines>,<mode>]** | Monitoring of log file. | **file** – full file name<br>**regexp** – regular expression for pattern<br>**encoding** - Code Page identifier<br>**maxlines** - Maximum number of new lines per second the agent will send to Zabbix Server or Proxy. This parameter overrides the 'MaxLinesPerSecond' option in zabbix_agentd.conf<br>**mode** - one of all (default), skip (skipping processing of older data)<br>Parameter **mode** will be supported from version 2.0. |

| | | |
|---|---|---|
| **logrt[file_format,<regexp>,<encoding>,<maxlines>,<mode>]** | Monitoring of log file with log rotation support. | **file_format** – full file name in format [absolute path][filename format as regexp]<br>**regexp** – regular expression for pattern<br>**encoding** - Code Page identifier<br>**maxlines** - Maximum number of new lines per second the agent will send to Zabbix Server or Proxy. This parameter overrides the 'MaxLinesPerSecond' option in zabbix_agentd.conf<br>**mode** - one of all (default), skip (skipping processing of older data)<br>Parameter **mode** will be supported from version 2.0. |
| **eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]** | Monitoring of event logs. | **name** – event log name<br>**regexp** – regular expression<br>**severity** – regular expression<br>The parameter accepts the following values: "Information", "Warning", "Error", "Failure Audit", "Success Audit"<br>**source** - Source identifier<br>**eventid** - regular expression<br>**maxlines** - Maximum number of new lines per second the agent will send to Zabbix Server or Proxy. This parameter overrides the 'MaxLinesPerSecond' option in zabbix_agentd.conf<br>**mode** - one of all (default), skip (skipping processing of older data)<br>Parameter **mode** will be supported from version 2.0. |
| **net.if.collisions[if]** | Out-of-window collision. | **if** - interface |

| | | |
|---|---|---|
| **net.if.in[if,<mode>]** | Network interface incoming statistic. | **if** - interface<br>**mode** –<br>**bytes** number of bytes (default)<br>**packets** number of packets<br>**errors** number of errors<br>**dropped** number of dropped packets |
| **net.if.list** | List of network interfaces: Type Status IPv4 Description | |
| **net.if.out[if,<mode>]** | Network interface outgoing statistic. | **if** - interface<br>**mode** –<br>**bytes** number of bytes (default)<br>**packets** number of packets<br>**errors** number of errors<br>**dropped** number of dropped packets |
| **net.if.total[if,<mode>]** | Sum of network interface incoming and outgoing statistics. | **if** - interface<br>**mode** –<br>**bytes** number of bytes (default)<br>**packets** number of packets<br>**errors** number of errors<br>**dropped** number of dropped packets |
| **net.tcp.dns[<ip>,zone]** | Checks if DNS service is up. | **ip** - IP address of DNS server (ignored)<br>**zone** - zone to test the DNS |
| **net.tcp.dns.query[<ip>,zone,<type>]** | Performs a query for the supplied DNS record type. | **ip** - IP address of DNS server (ignored)<br>**zone** - zone to test the DNS<br>**type** - Record type to be queried (default is SOA) |
| **net.tcp.listen[port]** | Checks if this port is in LISTEN state. | **port** - port number |

| | | |
|---|---|---|
| **net.tcp.port[<ip>,port]** | Check, if it is possible to make TCP connection to port number port. | **ip** - IP address(default is 127.0.0.1)<br>**port** - port number |
| **net.tcp.service[service,<ip>,<port>]** | Check if service is running and accepting TCP connections. | **service** - one of ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp<br>**ip** - IP address (default is 127.0.0.1)<br>**port** - port number (by default standard service port number is used) |
| **net.tcp.service.perf[service,<ip>,<port>]** | Check performance of service | **service** - one of ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp<br>ip - IP address (default is 127.0.0.1)<br>**port** - port number (by default standard service port number is used) |
| **proc.mem[<name>,<user>,<mode>,<cmdline>]** | Memory used by process name running under user user | **name** - process name<br>**user** - user name (default is all users)<br>**mode** - one of avg, max, min, sum (default)<br>**cmdline** - filter by command line |
| **proc.num[<name>,<user>,<state>,<cmdline>]** | Number of processes name having state running under user user | **name** - process name<br>**user** - user name (default is all users)<br>**state** - one of all (default), run, sleep, zomb<br>**cmdline** - filter by command line |
| **sensor[<temp>]** | Sensor reading. | **temp** - one of temp1, temp2, temp3. |
| **system.cpu.intr** | Device interrupts. | |
| **system.boottime** | Timestamp of system boot. | |

| | | |
|---|---|---|
| **system.cpu.load[<cpu>,<mode>]** | CPU load. | **cpu** - CPU number (default is all CPUs)<br>**mode** - one of avg1 (default),avg5 (average within 5 minutes), avg15 |
| **system.cpu.num[<type>]** | Number of CPUs. | **type** - one of online (default), max |
| **system.cpu.switches** | Context switches. | |
| **system.cpu.util[<cpu>,<type>,<mode>]** | CPU(s) utilisation. | **cpu** - CPU number (default is all CPUs)<br>**type** - one of idle, nice, user (default), system, kernel, iowait, interrupt, softirq, steal<br>**mode** - one of avg1 (default),avg5 (average within 5 minutes), avg15 |
| **system.run[command,<mode>]** | Run specified command on the host. | **command** - command for execution<br>**mode** - one of wait (default, wait end of execution), nowait (do no wait) |
| **system.hostname** | Return host name. | |
| **system.localtime** | System time. | **utc** - (default) the time since the Epoch (00:00:00 UTC, January 1, 1970), measured in seconds.<br>**local** - the time in the 'yyyy-mm-dd,hh:mm:ss.nn,+hh:mm' format<br>Parameters for this item will be supported from version 2.0. |
| **system.swap.in[<device>,<type>]** | Swap in. | **device** - swap device (default is all), **type** - one of count (default, number of swapins), pages (pages swapped in) |

| | | |
|---|---|---|
| **system.swap.out[<device>,<type>]** | Swap in. | **device** - swap device (default is all), **type** - one of count (default, number of swapouts), **pages** (pages swapped out) |
| **system.swap.size[<device>,<mode>]** | Swap space. | **device** - swap device (default is all), **type** - one of free (default, free swap space), total (total swap space), pfree (free swap space, percentage), pused (used swap space, percentage) |
| **system.uname** | Returns detailed host information. | |
| **system.uptime** | System's uptime in seconds. | |
| **system.users.num** | Number of users connected. | |
| **vfs.dev.read[device,<type>]** | Disk read statistics. | **device** - disk device (default is all)<br>**type** - one of sectors, operations, bytes, sps, ops, bps (must specify exactly which parameter to use, since defaults are different under various OSes) |
| **vfs.dev.write[device,<type>]** | Disk write statistics. | **device** - disk device (default is all)<br>**type** - one of sectors, operations, bytes, sps, ops, bps (must specify exactly which parameter to use, since defaults are different under various OSes) |
| **vfs.file.cksum[file]** | Calculate file check sum | **file** - full path to file |
| **vfs.file.exists[file]** | Check if file exists | **file** - full path to file |
| **vfs.file.md5sum[file]** | File's MD5 check sum | |
| **vfs.file.regexp[file,regexp,<encoding>]** | Find string in a file | **file** - full path to file, **regexp** - GNU regular expression<br>**encoding** - Code Page identifier |

| | | |
|---|---|---|
| **vfs.file.regmatch[file,regexp,<encoding>]** | Find string in a file | **file** - full path to file<br>**regexp** - GNU regular expression<br>**encoding** - Code Page identifier |
| **vfs.file.size[file]** | File size | **file** - full path to file |
| **vfs.file.time[file,<mode>]** | File time information. | **file** - full path to file<br>**mode** - one of modify (default, modification time), access - last access time, change - last change time |
| **vfs.fs.inode[fs,<mode>]** | Number of inodes | **fs** - filesystem<br>**mode** - one of total (default), free, used, pfree (free, percentage), pused (used, percentage) |
| **vfs.fs.size[fs,<mode>]** | Disk space | **fs** - filesystem<br>**mode** - one of total (default), free, used, pfree (free, percentage), pused (used, percentage) |
| **vm.memory.size[<mode>]** | Memory size | **mode** - one of total (default), shared, free, buffers, cached, pfree, available |
| **web.page.get[host,<path>,<port>]** | Get content of WEB page | **host** - hostname<br>**path** - path to HTML document (default is /)<br>**port** - port number (default is 80) |
| **web.page.perf[host,<path>,<port>]** | Get timing of loading full WEB page | **host** - hostname<br>**path** - path to HTML document (default is /)<br>**port** - port number (default is 80) |

| | | |
|---|---|---|
| **web.page.regexp[host,<path>,<port>,<regexp>,<length>]** | Get first occurrence of regexp in WEB page | **host** - hostname<br>**path** - path to HTML document (default is /)<br>**port** - port number (default is 80)<br>**regexp** - GNU regular expression, **length** - number of characters to return |

**Tabla 75. Claves de monitorización soportadas por el agente Zabbix**

## Comprobaciones sencillas (Simple checks)

Las comprobaciones sencillas habitualmente se utilizaran para monitorizar equipos que no tienen un agente Zabbix instalado o para chequeos remotos de servicios.

El listado de comprobaciones sencillas que soporta Zabbix se resume en la siguiente tabla:

| Key | Description | Return value |
|---|---|---|
| **ftp,<port>** | Checks if FTP server is running and accepting connections | 0 – FTP server is down<br>1 – FTP server is running<br>2 – timeout |
| **ftp_perf,<port>** | Checks if FTP server is running and accepting connections | 0 – FTP server is down<br>Otherwise, number of seconds spent connecting to FTP server. |
| **http,<port>** | Checks if HTTP server is running and accepting connections | 0 – HTTP server is down<br>1 – HTTP server is running<br>2 – timeout |
| **http_perf,<port>** | Checks if HTTP (WEB) server is running and accepting connections | 0 – HTTP (WEB) server is down<br>Otherwise, number of seconds spent connecting to HTTP server. |
| **icmpping[<target>,<packets>,<interval>,<size>,<timeout>]** | Checks if server is accessible by ICMP ping<br>**target** - host IP or DNS name<br>**packets** - number of packets<br>**interval** - time between successive packets in milliseconds<br>**size** - packet size in bytes<br>**timeout** - timeout in milliseconds | 0 – ICMP ping fails<br>1 – ICMP ping successful |

| | | |
|---|---|---|
| **icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>]** | Return percentage of lost packets<br>**target** - host IP or DNS name<br>**packets** - number of packets<br>**interval** - time between successive packets in milliseconds<br>**size** - packet size in bytes<br>**timeout** - timeout in milliseconds | Loss of packets in percents |
| **icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>]** | Return ICMP ping response time<br>**target** - host IP or DNS name<br>**packets** - number of packets<br>**interval** - time between successive packets in milliseconds<br>**size** - packet size in bytes<br>**timeout** - timeout in milliseconds<br>**mode** - one of min, max, avg (default) | Number of seconds |
| **imap,<port>** | Checks if IMAP server is running and accepting connections | 0 – IMAP server is down<br>1 – IMAP server is running<br>2 – timeout |
| **imap_perf,<port>** | Checks if IMAP server is running and accepting connections | 0 – IMAP server is down<br>Otherwise, number of seconds spent connecting to IMAP server. |
| **ldap,<port>** | Checks if LDAP server is running and accepting connections | 0 – LDAP server is down<br>1 – LDAP server is running<br>2 – timeout |
| **ldap_perf,<port>** | Checks if LDAP server is running and accepting connections | 0 – LDAP server is down<br>Otherwise, number of seconds spent connecting to LDAP server. |
| **nntp,<port>** | Checks if NNTP server is running and accepting connections | 0 – NNTP server is down<br>1 – NNTP server is running<br>2 – timeout |
| **nntp_perf,<port>** | Checks if NNTP server is running and accepting connections | 0 – NNTP server is down<br>Otherwise, number of seconds spent connecting to NNTP server. |
| **ntp,<port>** | Checks if NTP server is running and accepting connections | 0 – NTP server is down<br>1 – NTP server is running<br>2 – timeout |
| **ntp_perf,<port>** | Checks if NTP server is running and accepting connections | 0 – NTP server is down<br>Otherwise, number of seconds spent connecting to NTP server. |

| | | |
|---|---|---|
| **pop,<port>** | Checks if POP server is running and accepting connections | 0 – POP server is down<br>1 – POP server is running<br>2 – timeout |
| **pop_perf,<port>** | Checks if POP server is running and accepting connections | 0 – POP server is down<br>Otherwise, number of seconds spent connecting to POP server. |
| **smtp,<port>** | Checks if SMTP server is running and accepting connections | 0 – SMTP server is down<br>1 – SMTP server is running<br>2 – timeout |
| **smtp_perf,<port>** | Checks if SMTP server is running and accepting connections | 0 – SMTP server is down<br>Otherwise, number of seconds spent connecting to SMTP server. |
| **ssh,<port>** | Checks if SSH server is running and accepting connections | 0 – SSH server is down<br>1 – SSH server is running<br>2 – timeout |
| **ssh_perf,<port>** | Checks if SSH server is running and accepting connections | 0 – SSH server is down<br>Otherwise, number of seconds spent connecting to SSH server. |
| **tcp,port** | Checks if TCP service is running and accepting connections | 0 – TCP service is down<br>1 – TCP service is running<br>2 – timeout |
| **tcp_perf,port** | Checks if TCP service is running and accepting connections | 0 - the service on the port is down<br>Otherwise, number of seconds spent connecting to the TCP service. |

**Tabla 76. Comprobaciones simples soportadas por Zabbix**

## Comprobaciones internas (Internal checks)

Este tipo de comprobaciones permiten monitorizar aspectos internos de Zabbix. Las claves soportadas son:

| Key | Description | Comments |
|---|---|---|
| **zabbix[boottime]** | Startup time of Zabbix server process in seconds. | In seconds since the epoch. |
| **zabbix[history]** | Number of values stored in table HISTORY | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! |
| **zabbix[history_log]** | Number of values stored in table HISTORY_LOG | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! |

| | | |
|---|---|---|
| **zabbix[history_str]** | Number of values stored in table HISTORY_STR | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! |
| **zabbix[history_text]** | Number of values stored in table HISTORY_TEXT | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used!<br>**This item is supported starting from version 1.8.3.** |
| **zabbix[history_uint]** | Number of values stored in table HISTORY_UINT | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used!<br>**This item is supported starting from version 1.8.3.** |
| **zabbix[items]** | Number of items in Zabbix database | |
| **zabbix[items_unsupported]** | Number of unsupported items in Zabbix database | |
| **zabbix[log]** | Stores warning and error messages generated by Zabbix server. | Character. Add item with this key to have Zabbix internal messages stored. |
| **zabbix[proxy,<name>,<param>]** | Access to Proxy related information. | <name> - Proxy name<br>List of supported parameters (<param>):<br>lastaccess – timestamp of last heart beat message received from Proxy<br>For example, zabbix[proxy,"Germany",lastaccess]<br>Trigger function fuzzytime() can be used to check availability of proxies. |
| **zabbix[queue,<from>,<to>]** | Number of server monitored items in the Queue which are delayed by <from> to <to> seconds, inclusive. | <from> - default: 6 seconds<br><to> - default: infinity<br>Suffixes s,m,h,d,w are supported for these parameters.<br>**Parameters from and to are supported starting from version 1.8.3.** |
| **zabbix[trends]** | Number of values stored in table TRENDS | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! |

| | | |
|---|---|---|
| **zabbix[trends_uint]** | Number of values stored in table TRENDS_UINT | Do not use if MySQL InnoDB, Oracle or PostgreSQL is used!<br>**This item is supported starting from version 1.8.3.** |
| **zabbix[triggers]** | Number of triggers in Zabbix database | |
| **zabbix[uptime]** | Uptime of Zabbix server process in seconds. | |

# 14.3. ANEXO III. Macros implementadas en Zabbix

| MACRO | DESCRIPTION |
|---|---|
| {DATE} | Current date in yyyy.mm.dd. format. |
| {DISCOVERY.DEVICE.IPADDRESS} | IP address of the discovered device. |
| {DISCOVERY.DEVICE.STATUS} | Status of the discovered device. |
| {DISCOVERY.DEVICE.UPTIME} | Uptime of the discovered device. |
| {DISCOVERY.RULE.NAME} | Name of discovery rule. |
| {DISCOVERY.SERVICE.NAME} | Name of the service in which is based the discovery rule. |
| {DISCOVERY.SERVICE.PORT} | Source port. |
| {DISCOVERY.SERVICE.STATUS} | Status of the service in which is based the discovery rule. |
| {DISCOVERY.SERVICE.UPTIME} | Uptime of the service in which is based the discovey rule. |
| {ESC.HISTORY} | Escalation history. Log of previously sent messages. |
| {EVENT.ACK.HISTORY} | |
| {EVENT.ACK.STATUS} | |
| {EVENT.AGE} | Age of the event. Useful in escalated messages. |
| {EVENT.DATE} | Date of the event. |
| {EVENT.ID} | Numeric event ID which triggered this action. |
| {EVENT.TIME} | Time of the event. |
| {HOSTNAME<1-9>} | Host name of the Nth item of the trigger which caused a notification. |
| {HOST.CONN<1-9>} | IP and host DNS name depending on host settings. |
| {HOST.DNS<1-9>} | Host DNS name. |
| {IPADDRESS<1-9>} | IP address of the Nth item of the trigger which caused a notification. |

| | |
|---|---|
| **{ITEM.LASTVALUE<1-9>}** | The latest value of the Nth item of the trigger expression which caused a notification. Supported from Zabbix 1.4.3. It is alias to {{HOSTNAME}:{TRIGGER.KEY}.last(0)} |
| **{ITEM.LOG.AGE<1-9>}** | |
| **{ITEM.LOG.DATE<1-9>}** | |
| **{ITEM.LOG.EVENTID<1-9>}** | |
| **{ITEM.LOG.NSEVERITY<1-9>}** | |
| **{ITEM.LOG.SEVERITY<1-9>}** | |
| **{ITEM.LOG.SOURCE<1-9>}** | |
| **{ITEM.LOG.TIME<1-9>}** | |
| **{ITEM.NAME<1-9>}** | Name of the Nth item of the trigger which caused a notification. |
| **{ITEM.VALUE<1-9>}** | The latest value of Nth item of the trigger expression if used for displaying triggers. Historical (when event happened) value of Nth item of the trigger expression if used for displaying events. Supported from Zabbix 1.4.3. |
| **{NODE.ID<1-9>}** | |
| **{NODE.NAME<1-9>}** | |
| **{PROFILE.CONTACT<1-9>}** | Contact from host profile. |
| **{PROFILE.DEVICETYPE<1-9>}** | Device type from of host profile. |
| **{PROFILE.HARDWARE<1-9>}** | Hardware from host profile. |
| **{PROFILE.LOCATION<1-9>}** | Location from host profile. |
| **{PROFILE.MACADDRESS<1-9>}** | Mac Address from host profile. |
| **{PROFILE.NAME<1-9>}** | Name from host profile. |
| **{PROFILE.NOTES<1-9>}** | Notes from host profile. |
| **{PROFILE.OS<1-9>}** | OS from host profile. |
| **{PROFILE.SERIALNO<1-9>}** | Serial No from host profile. |
| **{PROFILE.SOFTWARE<1-9>}** | Sotware from host profile. |

| | |
|---|---|
| **{PROFILE.TAG<1-9>}** | Tag from host profile. |
| **{STATUS}** | Alias for {TRIGGER.STATUS}. |
| **{TIME}** | Current time in hh:mm.ss. |
| **{TRIGGER.COMMENT}** | Trigger comment. |
| **{TRIGGER.EVENTS.UNACK}** | Number of unacknowledged events for a map element in maps, or for the trigger which generated current event in notifications. Supported in map element labels since 1.8.3. |
| **{TRIGGER.EVENTS.PROBLEM.UNACK}** | Number of unacknowledged PROBLEM events for all triggers disregarding their state. Supported since 1.8.3. |
| **{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}** | Number of unacknowledged PROBLEM events for triggers in PROBLEM state. Supported since 1.8.3. |
| **{TRIGGER.EVENTS.ACK}** | Number of acknowledged events for a map element in maps, or for the trigger which generated current event in notifications. Supported since 1.8.3. |
| **{TRIGGER.EVENTS.PROBLEM.ACK}** | Number of acknowledged PROBLEM events for all triggers disregarding their state. Supported since 1.8.3. |
| **{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}** | Number of acknowledged PROBLEM events for triggers in PROBLEM state. Supported since 1.8.3. |
| **{TRIGGER.ID}** | Numeric trigger ID which triggered this action. |
| **{TRIGGER.KEY<1-9>}** | Key of the Nth item of the trigger which caused a notification. |
| **{TRIGGER.NAME}** | Name (description) of the trigger. |
| **{TRIGGER.NSEVERITY}** | Numerical trigger severity. Possible values: 0 - Not classified, 1 - Information, 2 - Warning,3 - Average, 4 - High, 5 - Disaster, Supported starting from Zabbix 1.6.2. |
| **{TRIGGER.SEVERITY}** | Trigger severity. Possible values: Not classified, Information, Warning, Average, High, Disaster, Unknown |
| **{TRIGGER.STATUS}, {STATUS}** | Trigger state. ON - if trigger is in TRUE state, OFF - if trigger is in FALSE state. **{STATUS} is deprecated.** |
| **{TRIGGER.URL}** | Trigger URL. |

| | |
|---|---|
| **{TRIGGER.VALUE}** | Current trigger value: 0 - trigger is in OFF state, 1 – trigger is in ON state, 2 – trigger UNKNOWN. This macro can also be used in trigger expressions. |
| **{TRIGGERS.UNACK}** | Number of unacknowledged triggers for a map element, disregarding trigger state. Trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged. |
| **{TRIGGERS.PROBLEM.UNACK}** | Number of unacknowledged PROBLEM triggers for a map element. Trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged. Supported since 1.8.3. |
| **{TRIGGERS.ACK}** | Number of acknowledged triggers for a map element, disregarding trigger state. Trigger is considered to be acknowledged if all of its PROBLEM events are acknowledged. Supported since 1.8.3. |
| **{TRIGGERS.PROBLEM.ACK}** | Number of acknowledged PROBLEM triggers for a map element. Trigger is considered to be acknowledged if all of its PROBLEM events are acknowledged. Supported since 1.8.3. |
| **{host:key.func(param)}** | Simple macros as used in trigger expressions. |
| **{$MACRO}** | Global and host level macros. |

**Tabla 77. Macros implementadas en Zabbix**

# 14.4. ANEXO IV. Mapeado de valores

En Zabbix es posible crear correspondencias entre valores numéricos y valores textuales para que así la comprensión de los datos de monitorización sea más sencilla.

Estos mapeados se configuran desde el menú *Administration -> General*, y, allí, seleccionando en el desplegable de la esquina superior derecha la entrada "*Value mapping*" podremos acceder a la funcionalidad que nos permite crearlos.

Los mapeados implementados en la plataforma de monitorización del sistema de videovigilancia son los siguientes:

| Nombre | Dominio de valores |
|---|---|
| **CPU Architecture** | 0 = x86 |
| | 1 = x86_64 |
| **Drive status** | 0 = Offline |
| | 1 = Offline JBOD |
| | 217 = Unconverted DCB |
| | 219 = Unsupported DCB |
| | 221 = DCB Data check |
| | 223 = DCB Orphan |
| | 236 = DCB Read Failure |
| | 238 = DCB Read timeout |
| | 241 = Drive timeout |
| | 245 = SMART failure |
| | 254 = Unsupported |
| | 255 = OK |
| **Host status [2]** | 0 = Unreachable |
| | 1 = Up |
| **RAID Unit status** | 0 = OK |
| | 1 = Verifying |
| | 2 = Initializing |
| | 3 = Degraded |
| | 4 = Rebuilding |
| | 5 = Recovering |
| | 6 = Migrating |
| | 7 = Inoperable |
| | 255 = Unknown |

| | |
|---|---|
| **RAID Unit Configuration** | 0 = RAID 0 |
| | 1 = RAID 1 |
| | 5 = RAID 5 |
| | 6 = RAID 6 |
| | 10 = RAID 10 |
| | 50 = RAID 50 |
| | 100 = RAID Subunit |
| | 102 = JBOD |
| | 104 = LUN |
| | 106 = IBOD |
| | 108 = CBOD |
| | 110 = SPARE |
| | 112 = DEAD |
| | 255 = Unknown |
| **Service state** | 0 = Down |
| | 1 = Up |
| **Windows Service state** | 0 = Running |
| | 1 = Paused |
| | 3 = Pause pending |
| | 4 = Continue pending |
| | 5 = Stop pending |
| | 6 = Stopped |
| | 7 = Unknown |
| | 255 = No such service |

**Tabla 78. Valores mapeados en la plataforma de monitorización**

## 14.5. ANEXO V. Ejemplo de fichero de configuración networker

```
#!/bin/sh
case "$1" in
    start)
        echo -n "Starting NetWorker daemons: "
        if [ -f /usr/sbin/nsrexecd ]; then
            echo -n " nsrexecd"
            #
            # Allow access from only the specified hosts.
            #
            /usr/sbin/nsrexecd -s nsrhost
        fi
        #
        # If lgtoserv is installed, start nsrd
        #
        if [ -f /usr/sbin/nsrd ]; then
            echo " nsrd"
            /usr/sbin/nsrd
        fi
        echo "."
        ;;
    stop)
        echo -n "Stopping NetWorker daemons:"
        if [ -f /usr/sbin/nsr_shutdown ]; then
            echo -n " nsr_shutdown -a -q"
            /usr/sbin/nsr_shutdown -a -q
        fi
        echo "."
        ;;
    restart|force-reload)
        $0 stop
        $0 start
        ;;
    *)
        echo "usage: `basename $0` {start|stop|restart|force-reload}"
        exit 1
        ;;
esac
exit 0
```

Donde "***nsrhost***" es el nombre del servidor de backup.

# 14.6. ANEXO VI. Gráficos de monitorización

## ❶ Servidores de grabación

Se incluyen aquí como ejemplo los gráficos creados para el servidor SERVERL01 (para el resto de servidores de grabación los gráficos son exactamente los mismos).

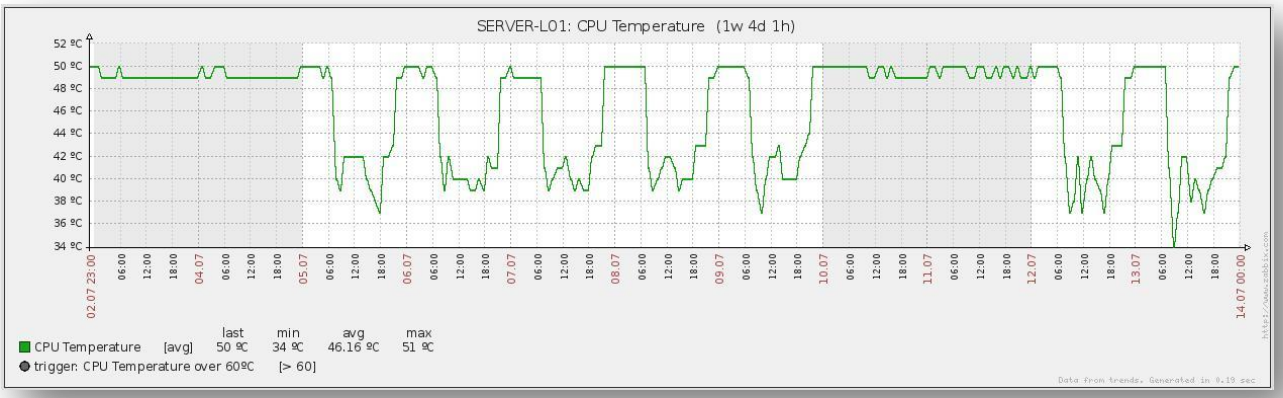



**Figura 45. Gráfico de temperatura de la CPU**

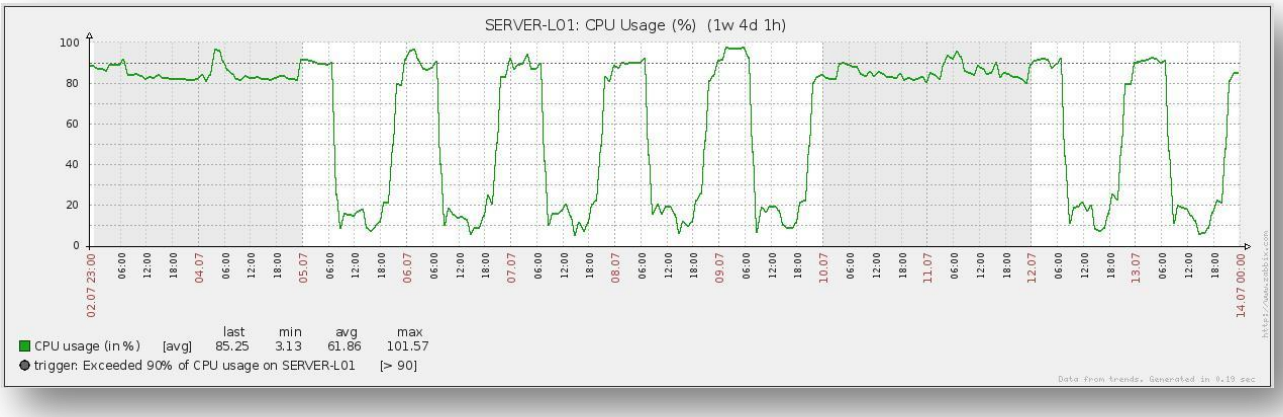



**Figura 46. Gráfico de uso de CPU**

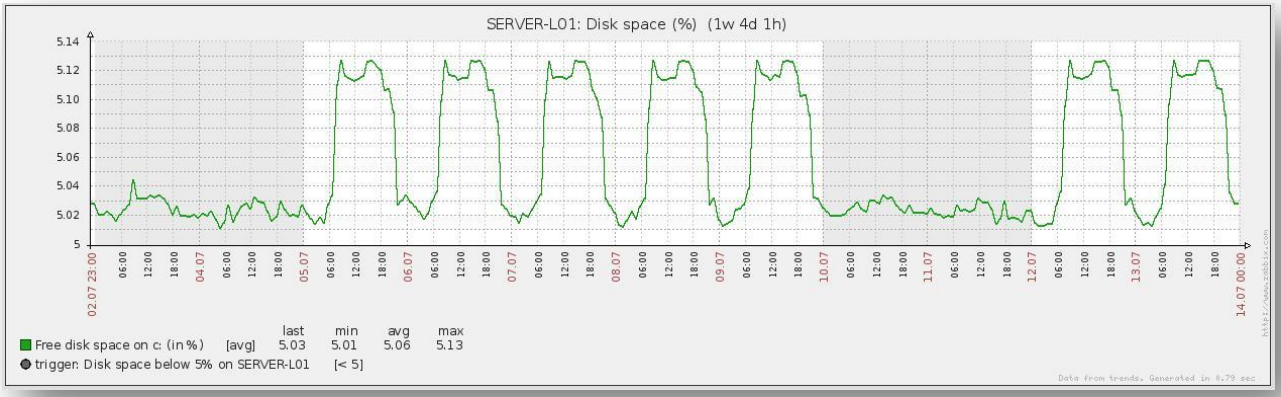



**Figura 47. Gráfico de espacio en disco**

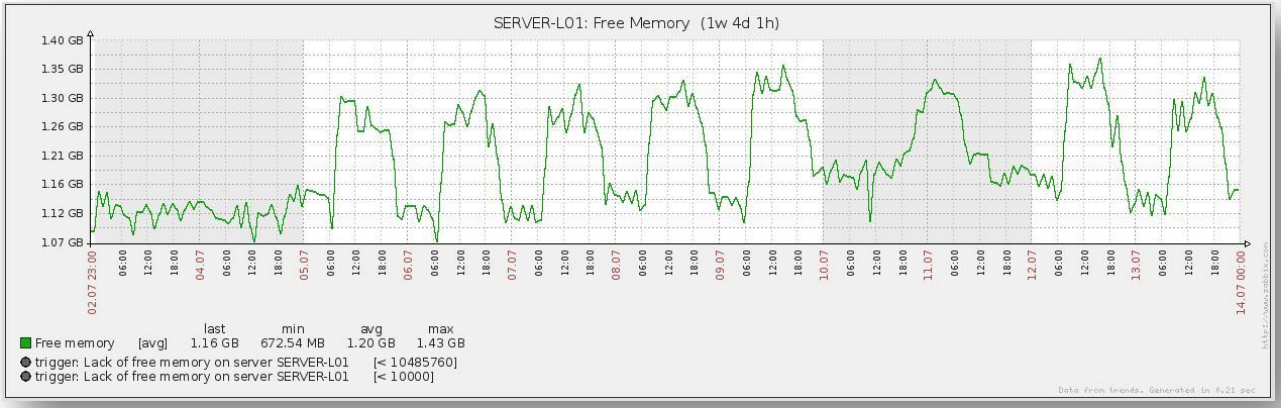



**Figura 48. Gráfico de memoria disponible**

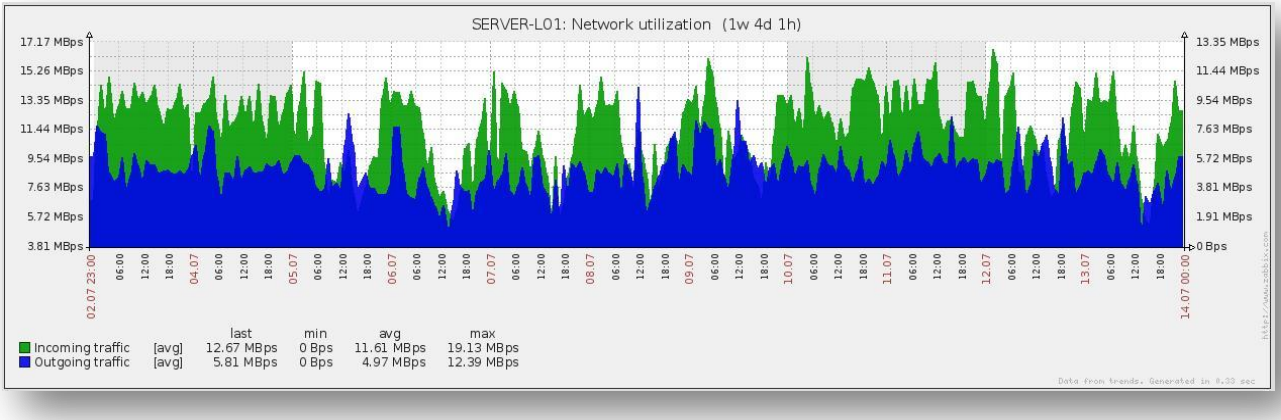



**Figura 49. Gráfico de tráfico de red**

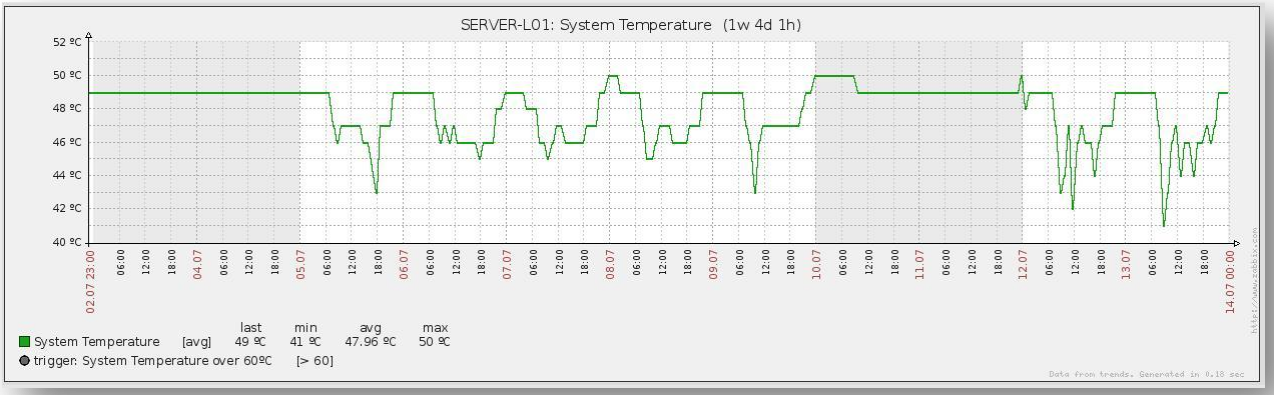



**Figura 50. Gráfico de temperatura del sistema**

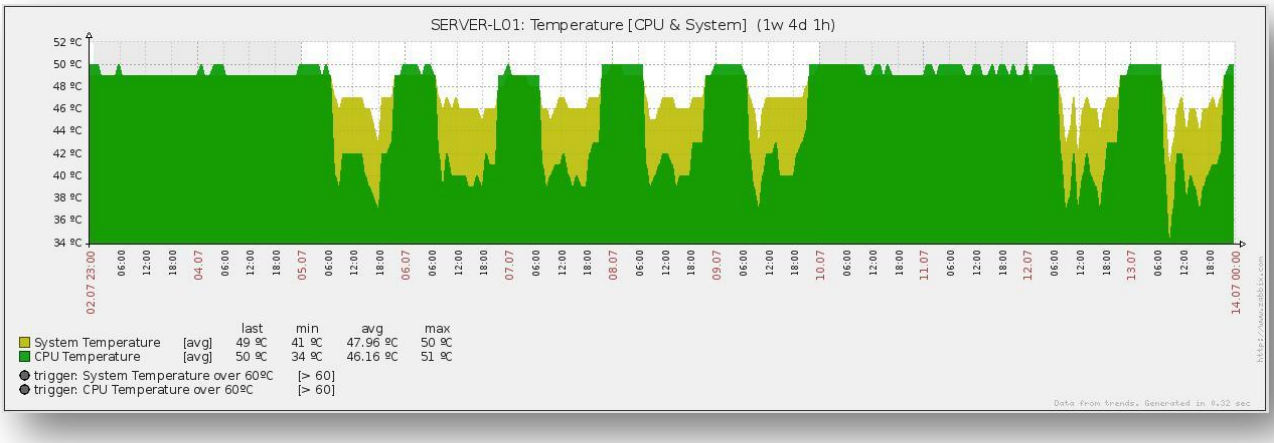



**Figura 51. Gráfico combinado de temperatura de CPU y de sistema**

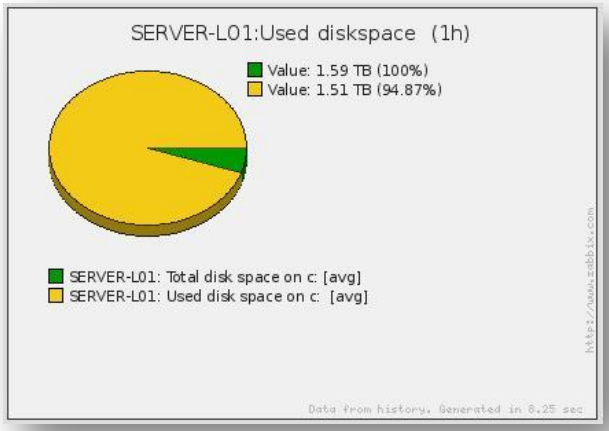



**Figura 52. Gráfico de porcentaje usado de espacio en disco**

## ❷ Servidor central Zabbix

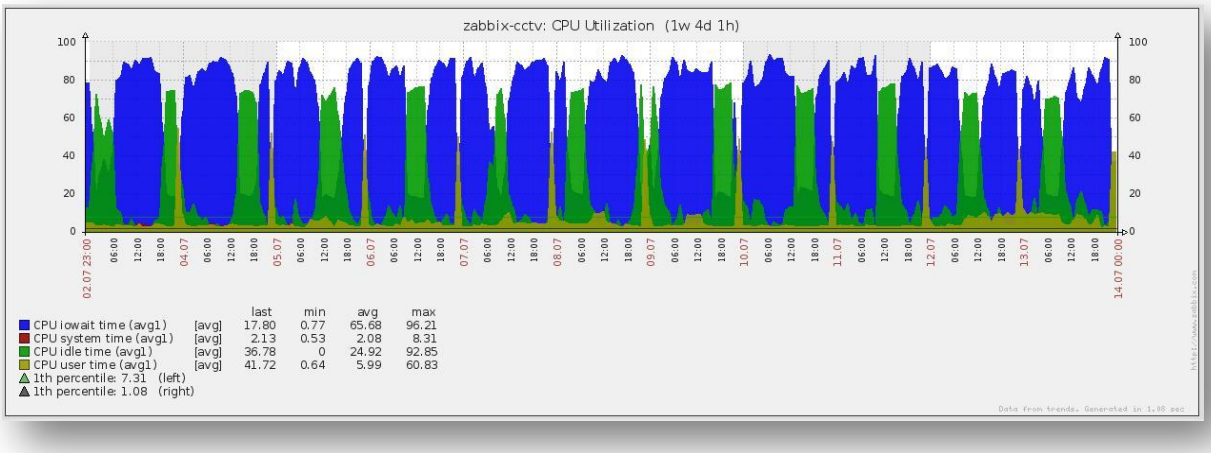



**Figura 53. Gráfico de utilización de CPU del servidor Zabbix**

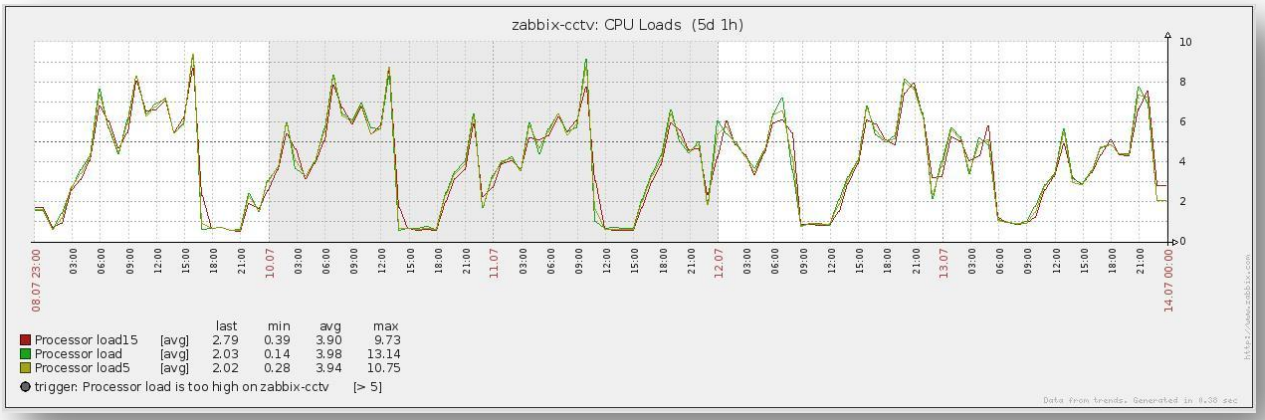



**Figura 54. Gráfico de carga de CPU en el servidor Zabbix**

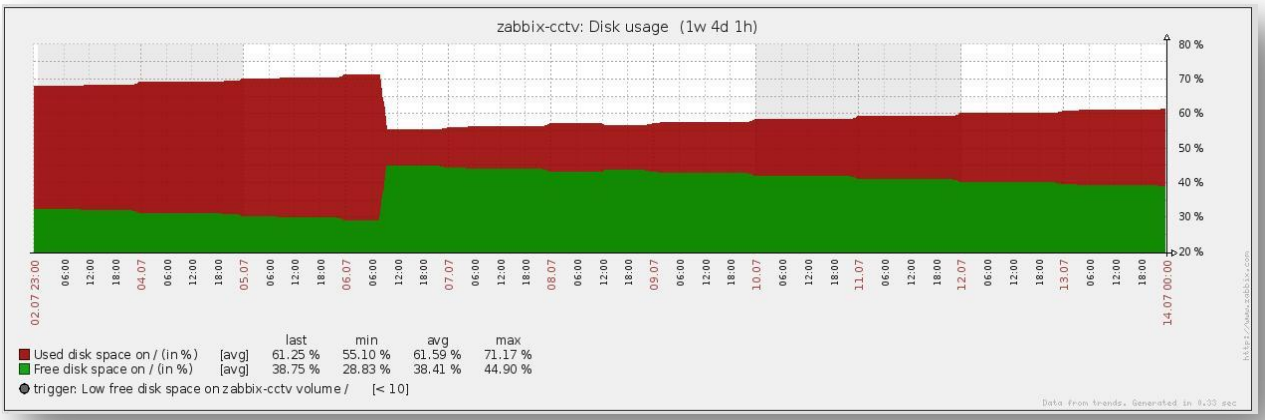



**Figura 55. Gráfico de uso de espacio en disco en el servidor Zabbix**

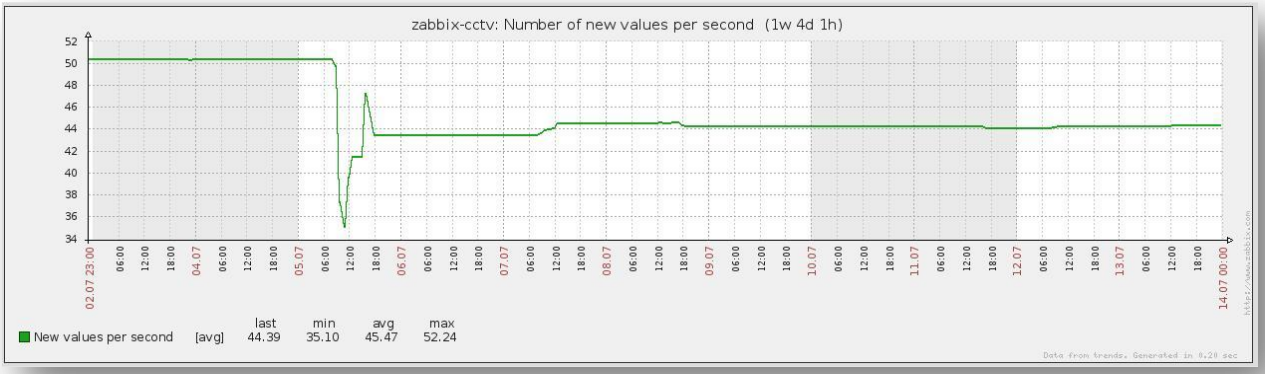



**Figura 56. Gráfico de nuevos valores por segundo en el servidor Zabbix**

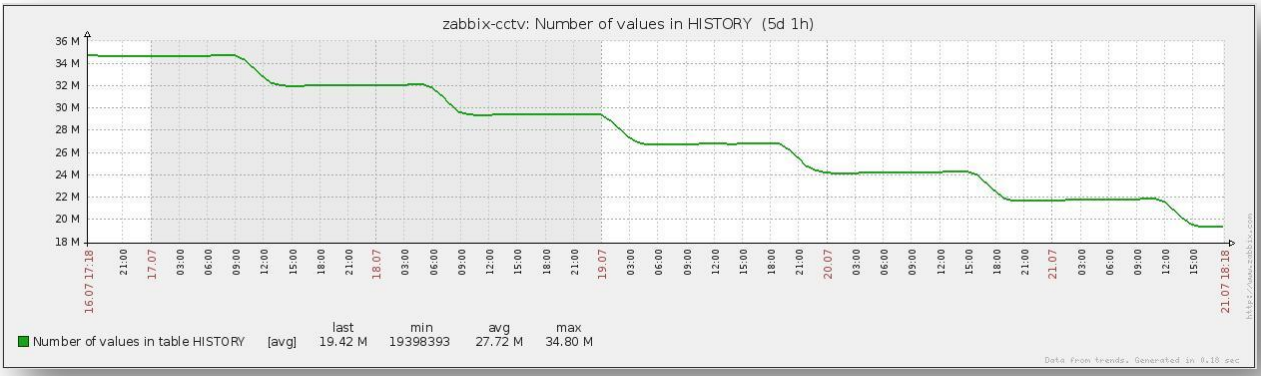



**Figura 57. Gráfico del número de valores en el historial del servidor Zabbix**

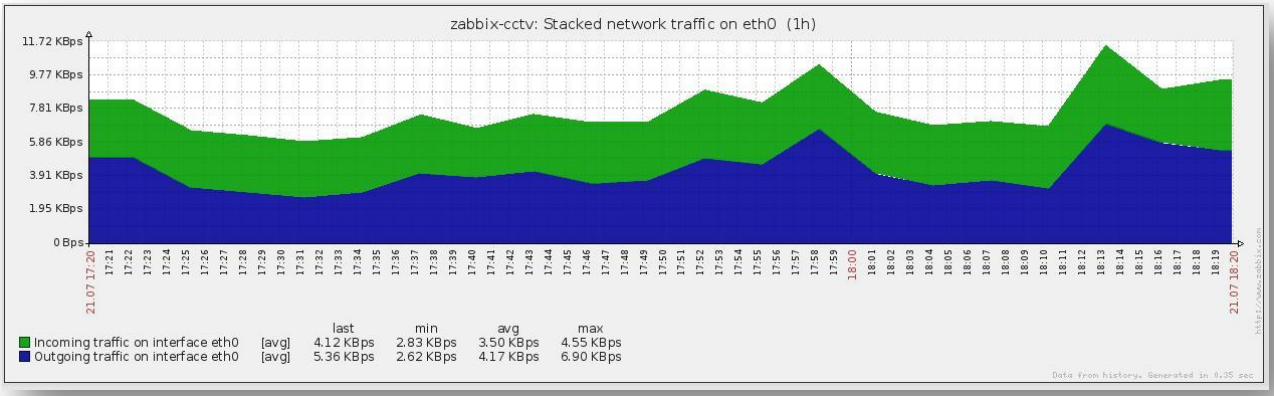



**Figura 58. Gráfico del tráfico de red en la interfaz eth0 del servidor Zabbix**

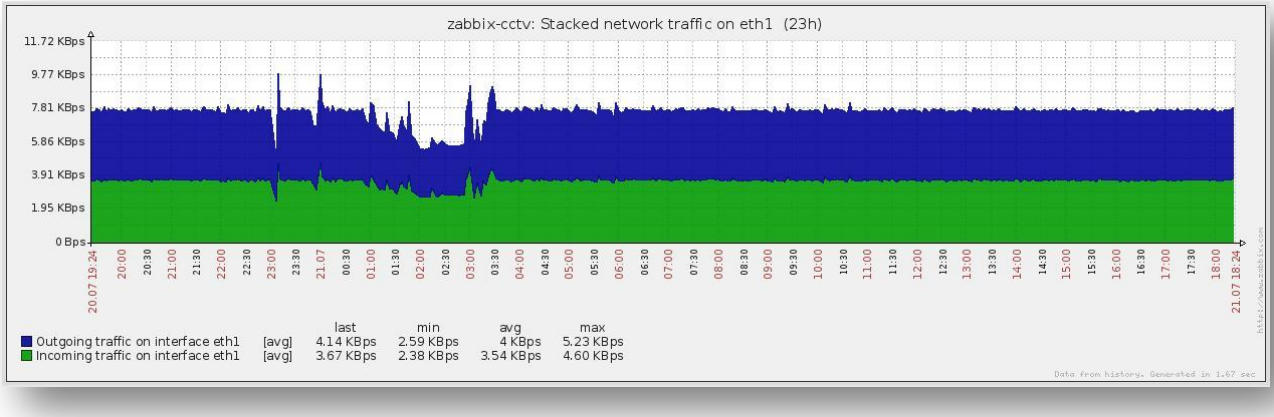



**Figura 59. Gráfico del tráfico de red en la interfaz eth1 del servidor Zabbix**

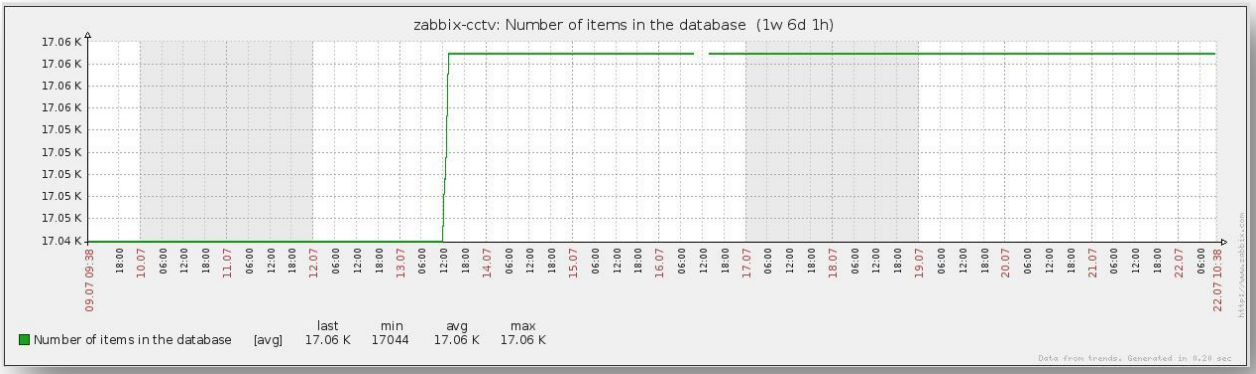



**Figura 60. Gráfico del número de items en el servidor Zabbix**

# 14.7. ANEXO VII. Sony RealShot Manager™

RealShot Manager™ es una aplicación software ideada para la monitorización de cámaras de red (*cámaras IP*) en un sistema de monitorización multipunto. Instalando este software en un equipo informático, tras unas sencillas configuraciones, es posible gestionar múltiples cámaras IP en una red, monitorizar imágenes, buscar y reproducir grabaciones, así como controlar cámaras entre otras funcionalidades.

Las principales características de este software creado por Sony son las siguientes:

- **Visualización simultánea de varias cámaras de red**: la pantalla principal de RealShot Manager permite visualizar imágenes de varias cámaras de red a la vez en el mismo área de pantalla. El diseño de las ventanas de monitorización de cámaras que se visualiza en la pantalla de RealShot Manager, así como el número, el tamaño o la distribución, se pueden configurar libremente según el objetivo y el entorno operativo. Además, se pueden importar datos ya existentes, como mapas y planos de planta, para después utilizarlos como fondo de la pantalla principal. También es posible, desde esa misma pantalla, controlar el comportamiento de la cámara, como la rotación, la inclinación y el zoom accediendo para ello a cada cámara de forma individual desde la ventana reservada a cada una de ellas en la pantalla principal de monitorización.
- **Grabación programada detallada y distintos modos de grabación**: es posible configurar un programa de grabación de imágenes para cada cámara o grupo de cámaras. De la misma manera, se ofrece la opción de configurar las cámaras para que comiencen a grabar cuando se active una alarma, cuyo origen puede ser la detección de movimiento o la señal proveniente de un sensor externo. Se dispone de una funcionalidad que permite grabar imágenes de forma manual desde la ventana de monitorización de cada cámara y, en el caso de desear reproducir las imágenes grabadas, existen una serie de controles al efecto.
- **Ajustes detallados de gestión y control de cámaras**: se pueden configurar grupos de cámaras para gestionarlas de forma efectiva, como uno para cada área o planta donde haya cámaras instaladas. Para cada cámara se pueden personalizar propiedades como la calidad y resolución de la imagen captada así como posiciones predefinidas de la propia cámara cuando sea necesario.
- **Detección de movimiento por software**: dado que el programa de grabación se puede configurar para modificar la detección de movimiento según la hora, es posible llevar a cabo varias operaciones, como ajustarlo para que cambie automáticamente la monitorización según sea de día o de noche. Sumado a lo anterior, el software RealShot Manager ofrece soporte para la detección de

movimiento basada en la propia cámara y funciones para la detección de objetos.

- **Funciones de filtrado utilizando metadatos de cámara**: el procesamiento preciso de la alarma es posible mediante el establecimiento de varios tipos de filtro y la utilización de los resultados del procesamiento de imagen enviados desde la cámara en forma de metadatos que contienen información de objetos. En tanto que el filtro puede aplicarse a los metadatos que ya han sido grabados, también es posible buscar áreas de interés una vez finalizada la grabación.
- **Soporte de los formatos JPEG y MPEG4** (la compatibilidad depende de qué tipos de formato de compresión soporta la cámara).
- **Monitorización, grabación y reproducción del sonido proveniente de una entrada de audio o del micrófono de la cámara.**
- **Configuración de la ubicación de almacenamiento de las grabaciones para cada cámara.**
- **Reproducción simultánea de varias grabaciones especificando una hora de inicio de la reproducción.**
- **Grabación por alarma utilizando el almacenamiento local de la cámara.** De este modo se garantiza la captura de imágenes estables y de alta calidad independientemente de las condiciones de la red.
- **Optimización automática de la base de datos sin interrupción del funcionamiento**, con lo cual se aseguran extensos periodos de utilización continuada.
- **Compatibilidad con la conexión y el uso de *"Generic Camera"*.**
- **Monitorización, grabación y reproducción hasta una resolución de 1280 x 960 píxeles.**

El diseño del sistema de videovigilancia actualmente implementado en la UC3M, desde el punto de vista del software y cámaras del fabricante Sony, es el siguiente:

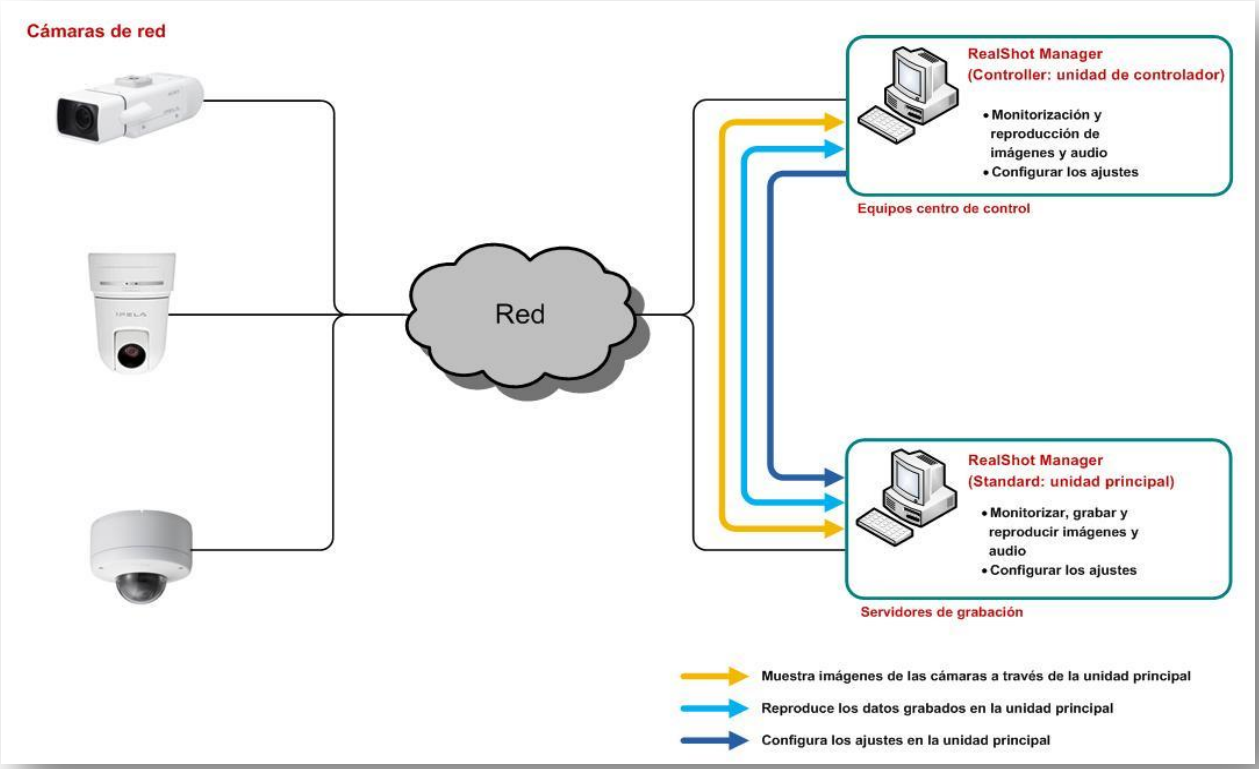



**Figura 61. Diseño del sistema Sony de videovigilancia**

En este diseño existen dos grupos bien diferenciados: las cámaras de red y los equipos en los que se encuentra instalado del software de gestión *RealShot Manager*. Existen dos tipos de instalación para el software según la función del equipo en el cual se realice: ***Standard*** y ***Controller***. En el modo *Standard*, el equipo actúa como un servidor en el sistema, por lo que se asigna esta instalación a los servidores de grabación del sistema de videovigilancia. En el caso de la opción *Controller*, el equipo funciona como cliente, por lo que este modo se reserva a los equipos de los centros de control desde los que se monitorizan las imágenes captadas por las cámaras y se reproducen las grabaciones almacenadas. Para realizar una instalación en modo *Controller* debe existir previamente en el sistema un equipo en el que se haya instalado previamente el software RealShot Manager en la modalidad *Standard*.

Así pues, el comportamiento de la relación "*Servidor de grabación-Equipo de centro de control*" se ajusta, como ya hemos dicho, a una arquitectura ***servidor-cliente*** en la que todas las operaciones que se realicen sobre las cámaras y sobre las imágenes monitorizadas y almacenadas deben pasar necesariamente por los servidores de grabación. Tal como se aprecia en la Figura 16, la monitorización y reproducción de grabaciones en los equipos de los centros de control así como los cambios en la configuración de las cámaras se debe realizar a través de una conexión con los

servidores de grabación. Para llevar a cabo cualquier tipo de cambio en los parámetros de las cámaras y en la monitorización de sus imágenes, ya sea en tiempo real o en forma de grabaciones, existen una serie de restricciones a nivel de privilegios de usuario de la aplicación y que trataremos posteriormente en la función ***"Gestión de usuarios"***. Todos los cambios que se efectúen desde el equipo cliente *Controller* implicarán las mismas modificaciones en los servidores *Standard*.

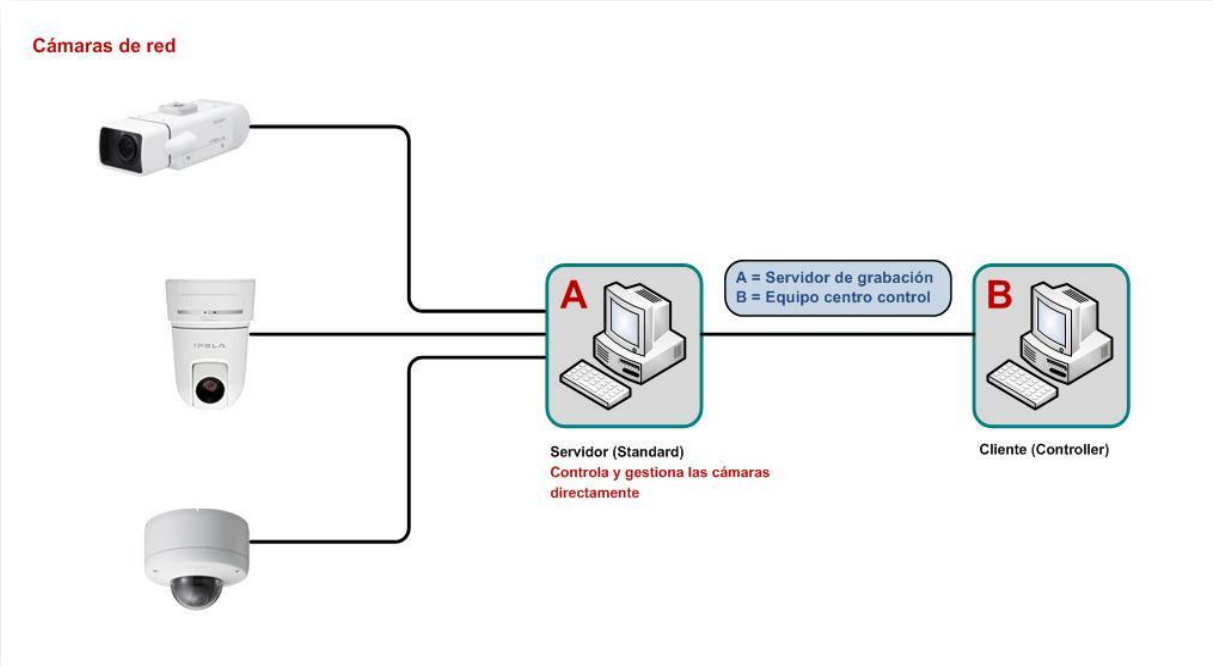



**Figura 62. Arquitectura cliente-servidor del sistema Sony**

El software *RealShot Manager* está equipado con una gran variedad de **funciones** que permiten, entre otras operaciones, configurar los ajustes en función del sistema operativo y de los objetivos perseguidos:

- Registro de dispositivos
- Ubicaciones de almacenamiento
- Diseño
- Monitorización
- Controles de la cámara
- Función de alarma
- Función de detección de movimiento
- Grabación
- Función de grabación por alarma utilizando el almacenamiento local de las cámaras

- Búsqueda y reproducción de los datos grabados
- Exportación de datos grabados
- Acciones
- Función de notificación por correo electrónico
- Gestión de usuarios
- Eliminación de grabaciones
- Servidores y clientes
- Generic Camera

Las funciones más importantes que trataremos son **Registro de dispositivos, Ubicaciones de almacenamiento, Servidores y clientes y, por último, Gestión de usuarios**.

a) **Registro de dispositivos**: con esta funcionalidad se registran dispositivos para poder gestionarlos con *RealShot Manager*, como pueden ser cámaras o controles de E/S. El **registro de una cámara** deberá hacerse desde el menú "*Mánager de configuración*". Allí, introduciremos en primer lugar la dirección IP de la cámara, el tipo de dispositivo (modelo de cámara en este caso) y un nombre con el que identificaremos la cámara. Una vez registremos la cámara, podemos proceder a la configuración de sus propiedades detalladas, entre las cuales se encuentran el usuario y contraseña necesarios para acceder a la cámara a través del explorador Web, los parámetros de la imagen y la detección de movimiento o el lugar donde se almacenarán las grabaciones de la propia cámara.

   En cuanto al **registro de un control de E/S**, se considerará previamente que las entradas de sensor se utilizan por lo general como activadores de acciones de alarma, como, por ejemplo una grabación por alarma en el equipo que tenga instalado *RealShot Manager*. Para el caso de las salidas, éstas se emplean para transmitir alarmas a los dispositivos equipados con funciones de entrada de alarma como una luz de advertencia o un mecanismo de apertura de puertas, por citar algunos ejemplos. Además, las salidas se pueden ejecutar manualmente desde la propia pantalla de *RealShot Manager*. Para definir un control de E/S la primera tarea es registrar el propio control, a continuación se configuran las propiedades de las *clavijas* y, por último, se asignan dichas clavijas a la cámara o cámaras correspondientes. Cada *clavija* es un identificador de la conexión entre el control de E/S y la cámara.

b) **Ubicaciones de almacenamiento**: este es uno de los aspectos más importantes en el sistema implementado actualmente. En todo sistema informático se debe contar con un espacio de almacenamiento, pues, en caso contrario, el sistema actuaría como una simple calculadora. Desde el punto de vista del sistema de videovigilancia, el almacenamiento juega un papel que, por sentido común, se

convierte en un punto especialmente crítico de cara al funcionamiento de dicho sistema: sin un lugar donde almacenar las imágenes que las cámaras han captado en momentos anteriores, básicamente no hay grabaciones; si tenemos un lugar donde almacenarlas, pero no hay espacio físico suficiente para ello, igualmente tampoco podremos disponer de ellas.

El software *RealShot Manager* permite configurar las propiedades para cada ubicación de almacenamiento que deseemos crear, como por ejemplo el tamaño máximo de los archivos correspondientes a las grabaciones. También es posible seleccionar una ubicación de almacenamiento personalizada para cada cámara. El primer paso es registrar la ubicación de almacenamiento a través del "***Mánager de configuración"***, siguiendo el mismo procedimiento que se sigue a la hora de registrar una cámara. Los elementos de la ubicación de almacenamiento susceptibles de ser configurados son, como se mencionó anteriormente, el tamaño máximo de archivo para cada grabación, el nombre y ruta de acceso a la ubicación, el modo de acceso remoto a las grabaciones (*Continuo*, cuando los archivos se transmiten al cliente desde el servidor; *Directo*, cuando el cliente, a través del sistema operativo, accede a la ubicación de almacenamiento del servidor donde están almacenadas las grabaciones y lee los archivos) y parámetros específicos para la conservación y borrado de grabaciones en los servidores.

Dentro de esos parámetros de borrado de grabaciones, se permite personalizar las condiciones bajo las cuales se ejecuta la limpieza de grabaciones antiguas. Así, se borrarán grabaciones cuando **la capacidad restante en el sistema de almacenamiento sea inferior a un cierto número** definido por el usuario. Otras opciones se reservan para decidir durante cuántos días se almacenarán las grabaciones antes de eliminarlas y cuál será la opción de borrado en ese caso. Es posible definir que se eliminen los archivos de grabaciones incluso cuando su antigüedad aún no llega al mínimo de días durante los cuales deben almacenarse para así garantizar una capacidad de almacenamiento constante. Para ello, se deberá activar la **sobreescritura de datos** dentro del mismo menú de la ubicación de almacenamiento.
Queda claro pues, la importancia que tiene la capacidad del almacenamiento no sólo para el sistema de videovigilancia sino también para el propio software de gestión. A modo de alerta, se pueden programar notificaciones en el software para que se envíen correos electrónicos en caso de que la ubicación de almacenamiento alcance el estado ***"DISK FULL"***, que es un límite en forma de porcentaje disponible sobre el total del espacio de almacenamiento.

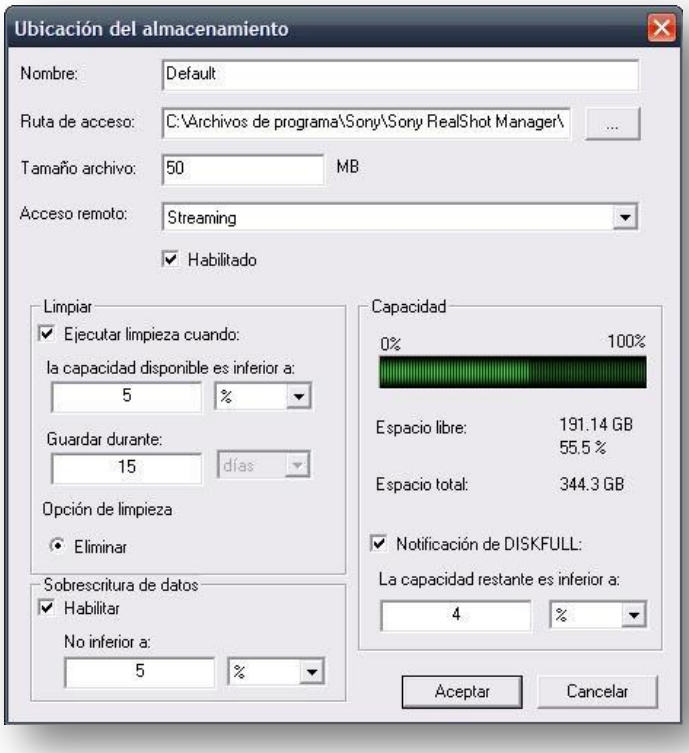



**Figura 63. Menú de configuración de la ubicación de almacenamiento**

Las ubicaciones de almacenamiento de los servidores de grabación corresponden al sistema RAID 5 implementado en cada uno de ellos y su configuración es la siguiente:

| | |
|---|---|
| **Nombre** | Default |
| **Ruta de acceso** | C:\Archivos de programa\Sony\Sony RealShot Manager\Recordings\ |
| **Tamaño máximo de archivo** | 50 MB |
| **Acceso remoto** | Habilitado mediante "Streaming" (modo *Continuo*) |
| **Ejecutar limpieza cuando** | La capacidad disponible es inferior al 5% |
| | Guardar grabaciones durante 15 días |
| | Opción de limpieza "eliminar" activada |
| **Sobreescritura de datos** | Habilitada, no inferior al 5% |
| **Notificación de DISKFULL** | Cuando la capacidad restante es inferior al 4% |

**Tabla 79. Configuración del almacenamiento en los servidores de grabación**

c) **Servidores y clientes**: anteriormente se mencionó que la arquitectura del sistema implementado es del tipo "*cliente-servidor*". Es necesaria una serie de ajustes para cada equipo dentro de un sistema que responda a dicha arquitectura. Si la instalación del equipo es de tipo *Standard*, el equipo

funcionará como servidor y, para que los clientes lo utilicen, deberán configurarse sus propiedades de compartición. Lo mismo sucede para el caso de las cámaras, pues de no habilitarse la opción que permite compartirlas, no será posible acceder a ellas para visualizar/reproducir las imágenes.

En el caso opuesto, versión *Controller*, el equipo funciona como cliente en el sistema y, para utilizarlo como controlador, se deberá registrar y compartir previamente un equipo que haga las veces de servidor.

d) **Gestión de usuarios**: se dispone de funciones de seguridad avanzadas con el fin de gestionar usuarios múltiples. Así, es posible crear tanto usuarios individuales como grupos de usuarios y definir, a nivel de usuario o de grupo, qué funciones están disponibles. Los privilegios a asignar van desde el control de entradas y salidas de las cámaras, la configuración de las propiedades de las mismas, la búsqueda y reproducción de grabaciones hasta operar con el movimiento de las cámaras, por citar algunas.

   Los equipos que cuentan con instalación de tipo *Controller*, ubicados en el centro de control, son operados por el personal de seguridad física de la UC3M. Para ellos, se definen dos niveles distintos de usuarios en lo que a privilegios se refiere y que se traducen en la creación de dos grupos de usuarios diferenciados en *RealShot Manager*: *Operador* y *Jefe de equipo*. El perfil de jefe de equipo dispone de acceso a las mismas funcionalidades que el operador y, sobre éstas, añade la búsqueda y reproducción de grabaciones almacenadas en los servidores.

   Además de estas dos categorías de usuario, existe un usuario administrador de la aplicación con permisos para la definición de grupos de usuarios, inclusión de usuarios individuales, definición de privilegios por usuario/grupo y configuración del acceso del usuario a grupos de cámaras específicas. Junto a todas ellas, el administrador podrá editar planes de trabajo de las cámaras, configurar las entradas y salidas o administrar el audio, entre otras funciones.

# 14.8. ANEXO VIII. Especificaciones de las cámaras de videovigilancia

Las especificaciones detalladas en las siguientes tablas se han extraído de la página oficial del fabricante Sony **[22]:**

- **Modelo SNC-CS50P** [23]

| **Imagen** | |
|---|---|
| Sensor | Sensor CCD de 1/3" (tecnología SuperExwave) |
| Número de píxeles efectivos | NTSC: 380.000 píxeles, 768 (H) x 494 (V); PAL: 440.000 píxeles, 752 (H) x 582 (V) |
| Objetivo | Varifocal, de montura CS y 2,7x |
| Distancia focal | f=2.9 - 8 mm |
| Ángulo horizontal de visualización | de 94 a 35° |
| Número F | F 0.95 (extremo macro), F1.6 (extremo tele) |
| Iris | Auto/Manual (F0.95 - cerrado) |
| Velocidad del obturador | de 1/60 (1/50) a 1/10.000 segundos |
| Distancia mínima al objeto | 300 mm |
| Otras funciones | Día/Noche, Detección de objetos, Estabilizador de imagen, Protección contra la manipulación de imágenes |

| **Cámara** | |
|---|---|
| Frecuencia de cuadro máxima | JPEG: modo VGA, máx.de 30 fps (NTSC), 25 fps (PAL); modo QVGA, máx. de 30 fps (NTSC), 25 fps (PAL)<br>MPEG4: modo VGA, máx. de 30 fps (NTSC), 25 fps (PAL); modo QVGA, máx. de 30 fps (NTSC), 25 fps (PAL)<br>H.264: modo VGA, máx. de 10 fps (NTSC), 8 fps (PAL); modo QVGA: máx. de 30 fps (NTSC), 25 fps (PAL) |
| Tamaño de las imágenes | 640 x 480(VGA), 320 x 240(QVGA), 160 x 120(QQVGA) |
| Formato de compresión | JPEG, MPEG4, H.264 (seleccionable) |

| **Audio** | |
|---|---|
| Formato de compresión | Entrada de audio: G.711 (64 kbps), G.726(40, 32, 24, 16 kbps) Salida de audio: G.711 (64 kbps), G.726(40, 32, 24, 16 kbps) |

| **Red** | |
|---|---|
| Protocolos | TCP/IP, ARP, ICMP, HTTP, FTP (cliente/servidor), SMTP, DHCP, DNS, NTP, RTP/RTCP, SNMP (MIB-2) |
| Acceso simultáneo máximo | 20 usuarios |

| **Interfaz** | |
|---|---|
| Ethernet | 10BASE-T / 100BASE-TX (RJ-45) |
| Interfaz serie | RS-232C |
| Puerto de E/S | Entradas de sensor x 2, Salidas de alarma x2 |
| Ranura de tarjetas | Tarjeta PC (Tipo II) |
| Salida de vídeo analógica | Vídeo compuesto (1Vp-p) |

| | |
|---|---|
| Entrada de micrófono externo | Mini-jack (Monaural), 2,2 k ohm, Alimentación 2,5 V CC |
| Salida de audio | Mini-jack (Monaural), Nivel máx. de salida: 1 Vrms |

| **Salida de vídeo analógica** | |
|---|---|
| Sistema de señal | NTSC/PAL |
| Resolución horizontal | 540 líneas (NTSC) 540 líneas (PAL) |
| Relación señal-ruido | Más de 50 dB |

| **Requisitos del sistema** | |
|---|---|
| Sistema operativo | Windows 2000/ Windows XP |
| CPU | Pentium4 1,5 GHz como mínimo (mínimo de 2,4 GHz recomendable) |
| Explorador de web | Microsoft Internet Explorer Ver6.0 o superior |

| **General** | |
|---|---|
| Peso | 750 g aprox. (sin cubierta delantera ni trasera) 880 g aprox. (con cubiertas delanteras y trasera) |
| Dimensiones en mm (An. x Alt. x Prof.) | 84 x 69 x 196 (sin cubiertas) 96,7 x 69 x 265 (con cubiertas) |
| Requisitos de alimentación | PoE (IEEE-802.3af), 24 V CA, 12 V CC |
| Consumo | 9 W máx. |
| Temperatura de funcionamiento | de 0 a +50 °C |
| Temperatura de almacenamiento | de -20 a +60 °C |
| Humedad de funcionamiento | de 20 a 80% |
| Humedad de almacenamiento | de 20 a 95% |

- **Modelo SNC-DF80P [24]**

| **Imagen** | |
|---|---|
| Tamaño de imagen (H x V) | 640 x 480(VGA), 320 x 240(QVGA), 160 x 120(QQVGA) |
| Formato de compresión | JPEG, MPEG4, H.264 (seleccionable) |
| Frecuencia de cuadro máxima | JPEG: modo VGA, máx, 25fps (PAL); modo QVGA: máx, 25fps (PAL)<br>MPEG4: modo VGA, 25fps (PAL); modo QVGA: máx, 25fps (PAL)<br>H.264: modo VGA, máx. 8fps (PAL); modo QVGA: máx, 25fps (PAL) |

| **Cámara** | |
|---|---|
| Dispositivo de captación de imagen | Sensor CCD de 1/3" (tecnología SuperExwave) |
| Número de píxeles efectivos (H x V) | 440.000 píxeles, 752 (H) x 582 (V) |
| Obturador lento | 1/50 a 1/10.000 s |
| Relación de zoom | Óptica x3.6 ( x1.5 Zoom Digital) |
| Distancia focal | f = 2,8 - 10 mm |
| Ángulo horizontal de visualización | de 100.8° a 27.7° |

| | |
|---|---|
| Número F | F 3 (extremo macro), F3.0 (extremo tele) |
| Iris | Automático |
| Iluminación mínima | @50IRE : 0.6lx (Color), 0.06 lx (B/N) (AGC ON, F1.3)@30IRE : 0.35lx (Color), 0.03lx (B/N)(AGC ON, F1.3) |
| Distancia mínima al objeto | 300 mm |
| Otras funciones | Día/Noche, integración dinámica de cuadros, estabilizador de la imagen, enmascaramiento de zona privada |

| **Red** | |
|---|---|
| Protocolos | IPv4, TCP, UDP, ARP, ICMP, IGMP, HTTP, FTP (servidor/cliente), SMTP, DHCP, DNS, NTP, RTP/RTCP, RTSP, SNMP (MIB-2) |
| Acceso simultáneo máximo | 20 usuarios |

| **Interfaz** | |
|---|---|
| Ethernet | 10BASE-T / 100BASE-TX (RJ-45) |
| Puerto de E/S | Entradas de sensor x 2, Salidas de alarma x2 |
| Salida de vídeo compuesto analógica | Vídeo compuesto (1Vp-p) |
| Entrada de micrófono externo | Mini-jack (Monaural), 2,2 k ohm, Alimentación 2,5 V CC |
| Salida de audio | Mini-jack (Monaural), Nivel máx. de salida: 1 Vrms |

| **Salida de vídeo compuesto analógica** | |
|---|---|
| Sistema de señal | PAL |
| Resolución horizontal | 540 líneas de TV |
| Relación señal-ruido | Más de 50 dB |

| **Requisitos del sistema** | |
|---|---|
| Sistema operativo | Windows 2000/ Windows XP |
| CPU | Pentium4 1,5 GHz como mínimo (mínimo de 2,4 GHz recomendable) |
| Explorador de web | Microsoft Internet Explorer Ver6.0 o superior |

| **General** | |
|---|---|
| Peso | Aprox. 1.8 kg |
| Dimensiones en mm (An. x Alt. x Prof.) | 177.5x 141.5 (V x H mm) |
| Requisitos de alimentación | PoE (IEEE-802.3af), 24 V CA, 12 V CC |
| Consumo | 10 W máx. |
| Temperatura de funcionamiento | de -10 °C a 50 °C |
| Temperatura de almacenamiento | de -20 °C a 60 °C |
| Humedad de funcionamiento | de 20 a 80% (Sin condensación) |
| Humedad de almacenamiento | de 20 a 95% (Sin condensación) |

- **Modelo SNC-RX550P [25]**

| **Imagen** | |
|---|---|
| Tamaño de imagen (H x V) | 640 x 480(VGA), 320 x 240(QVGA), 160 x 120(JPEG, MPEG-4.H.264) |
| Formato de compresión | JPEG, MPEG4, H.264 |
| Frecuencia de cuadro máxima | JPEG-MPEG4: máx, 25fps (VGA)<br>H.264: máx, 8 fps (VGA) |

| **Cámara** | |
|---|---|
| Dispositivo de captación de imagen | Sensor CCD de 1/4" (tecnología SuperExwave) |
| Número de píxeles efectivos (H x V) | 440.000 píxeles, 752 (H) x 582 (V) |
| Obturador lento | 1 a 1/10.000 s |
| Relación de zoom | Óptica x26 ( x312 Zoom Digital) |
| Distancia focal | f = 3,5 - 91 mm |
| Ángulo horizontal de visualización | de 2.2° a 54.2° |
| Número F | F 1.6 (extremo macro), F3.8 (extremo tele) |
| Iris | Automático |
| Iluminación mínima | @50IRE : 0.6lx (Color), 0.06 lx (B/N) (AGC ON, F1.3)@30IRE : 0.35lx (Color), 0.03lx (B/N)(AGC ON, F1.3) |
| Distancia mínima al objeto | 320 mm (macro), 1500 mm (tele) |
| Ángulo de giro | 360° ininterrumpidos |
| Velocidad de giro | 300° máx. |
| Ángulo de inclinación | -90° a 0° |
| Velocidad de inclinación | 300° máx. |
| Otras funciones | Día/Noche, integración dinámica de cuadros, estabilizador de la imagen, enmascaramiento de zona privada, presets de posición, detección de objetos desatendida, detección de movimiento avanzada |

| **Red** | |
|---|---|
| Protocolos | TCP/IP, HTTP, ARP, ICMP, FTP, SMTP, DHCP, SNMP, DNS, NTP |
| Acceso simultáneo máximo | 20 usuarios |

| **Interfaz** | |
|---|---|
| Ethernet | 10BASE-T / 100BASE-TX (RJ-45) |
| Puerto de E/S | Entradas de sensor x 2, Salidas de alarma x2 |
| Interfaz serie | RS-232C |
| Ranura de tarjetas | Tarjeta PC (Tipo II), Memory Stick |
| Salida de vídeo compuesto analógica | Vídeo compuesto (1Vp-p) |
| Entrada de micrófono externo | Mini-jack (Monaural), 2,2 k ohm, Alimentación 2,5 V CC |
| Salida de audio | Mini-jack (Monaural), Nivel máx. de salida: 1 Vrms |

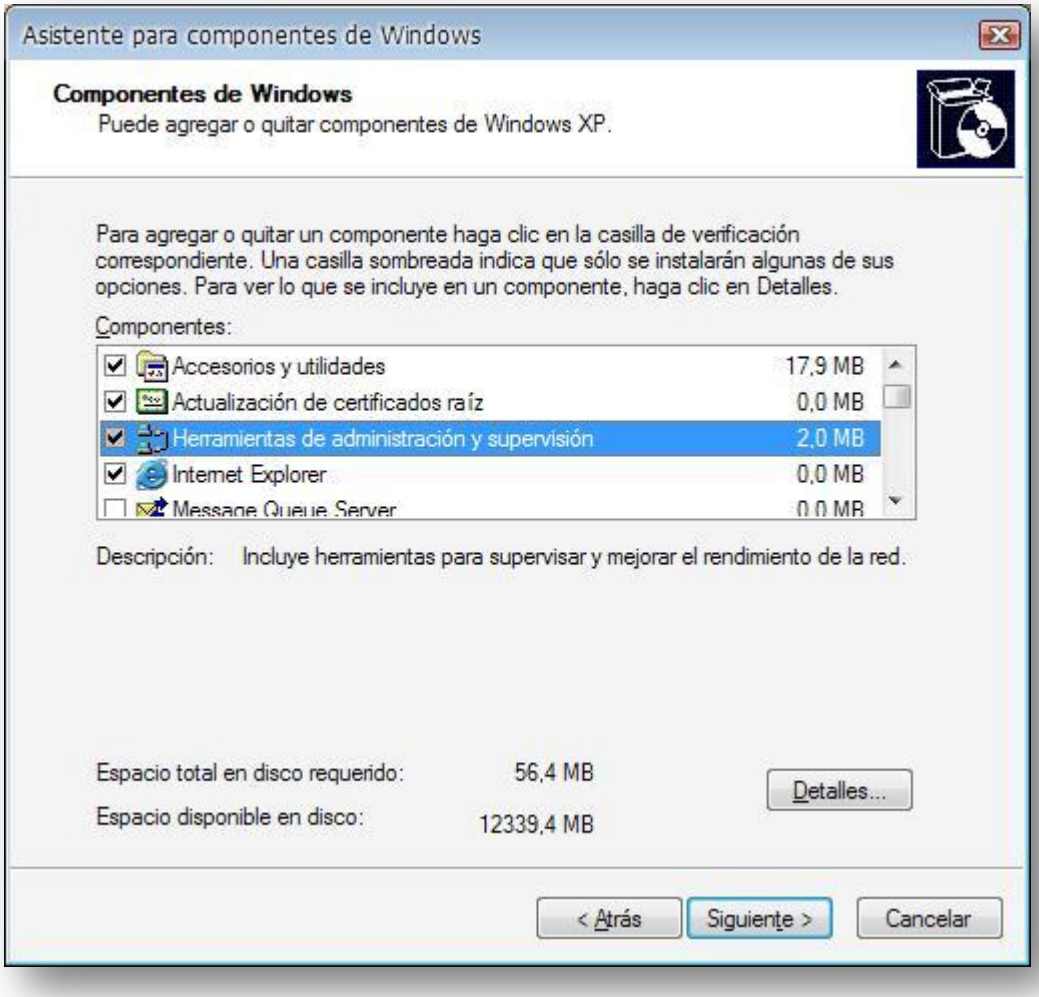
| **Salida de vídeo analógica** | |
|---|---|
| Sistema de señal | PAL |
| Resolución horizontal | 460 líneas de TV |
| Relación señal-ruido | 50 dB |

| **Requisitos del sistema** | |
|---|---|
| Sistema operativo | Windows 2000/ Windows XP |
| CPU | Pentium4 1,5 GHz como mínimo (mínimo de 2,4 GHz recomendable) |
| Explorador de web | Microsoft Internet Explorer Ver6.0 o superior |

| **General** | |
|---|---|
| Peso | Aprox. 2.2 kg |
| Dimensiones en mm (An. x Alt. x Prof.) | 160 x 160 x 230 mm |
| Requisitos de alimentación | PoE (IEEE-802.3af), 24 V CA, 12 V CC |
| Consumo | 25 W máx. |
| Temperatura de funcionamiento | de 0° °C a 50 °C |
| Temperatura de almacenamiento | de -20 °C a 60 °C |
| Humedad de funcionamiento | de 20 a 80% (Sin condensación) |
| Humedad de almacenamiento | de 20 a 95% (Sin condensación) |

# 14.9. ANEXO IX. Guía de instalación de SNMP en Windows

Desde el **Panel de control** de Windows, seleccionamos **Añadir o quitar programas** y, dentro de esta entrada, escogeremos la opción **Añadir o quitar componentes de Windows.** En ese momento aparecerá la siguiente ventana:




Seleccionamos la opción marcada y pulsamos en **Detalles:**

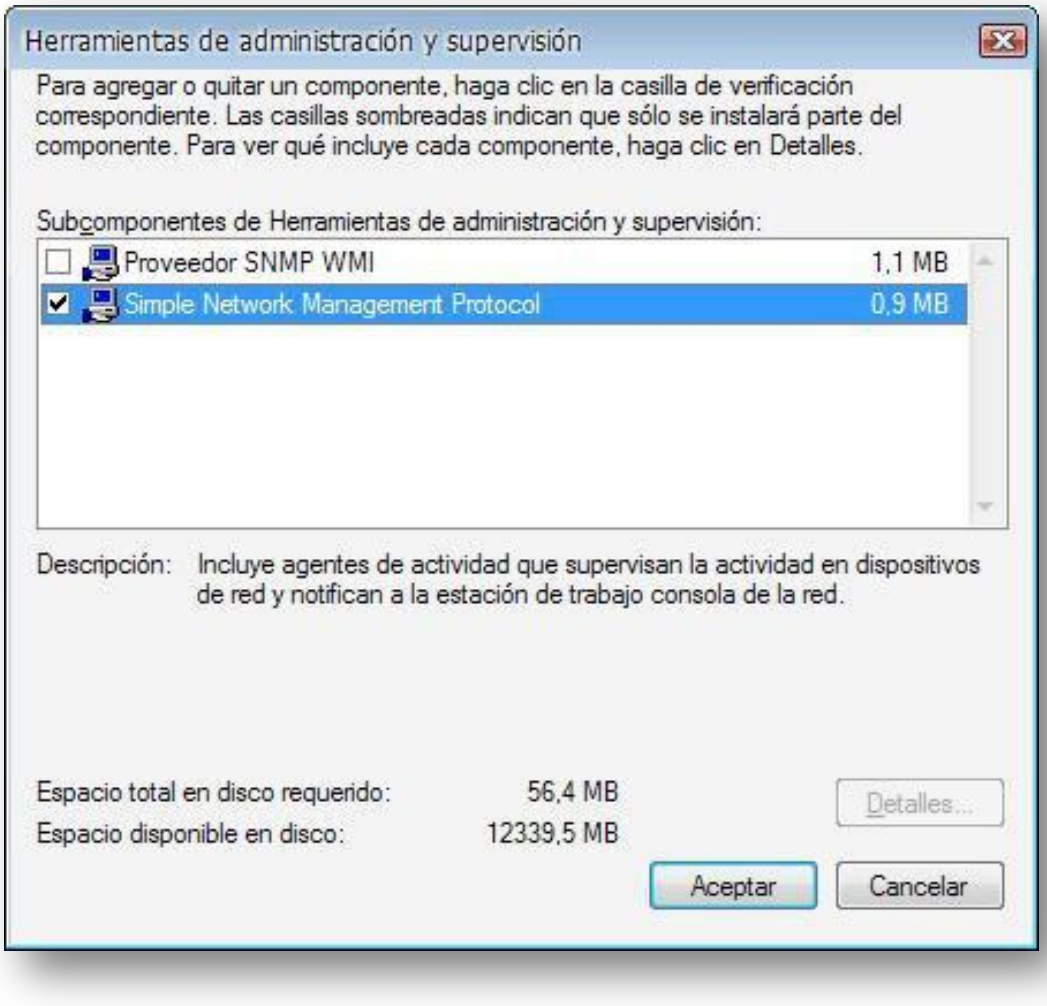



**Figura 64. Instalación de SNMP en Windows**

Seleccionamos el protocolo SNMP y pulsamos en **Aceptar**. Se procederá entonces a la copia de archivos necesarios para la instalación.

Tras la instalación del protocolo, accedemos a las **Herramientas administrativas** para configurar el servicio SNMP. En las propiedades del servicio aparecerán varias pestañas, de las cuales prestaremos atención a **Seguridad**. Desde ésta estableceremos cómo puede ser accedida la información del equipo a través de SNMP. El agente SNMP de Windows ofrece soporte para las versiones SNMPv1 y SNMPv2c, por lo que el modelo de seguridad está basado en el string de la comunidad y direcciones IP para la autenticación. Además, el agente puede aceptar consultas SNMP desde todos los equipos o hosts o sólo desde una serie de equipos determinados y se pueden configurar una o más comunidades para lectura y escritura de información. Por defecto, sólo se aceptan con permisos de sólo lectura las peticiones con el string de la comunidad *public*.

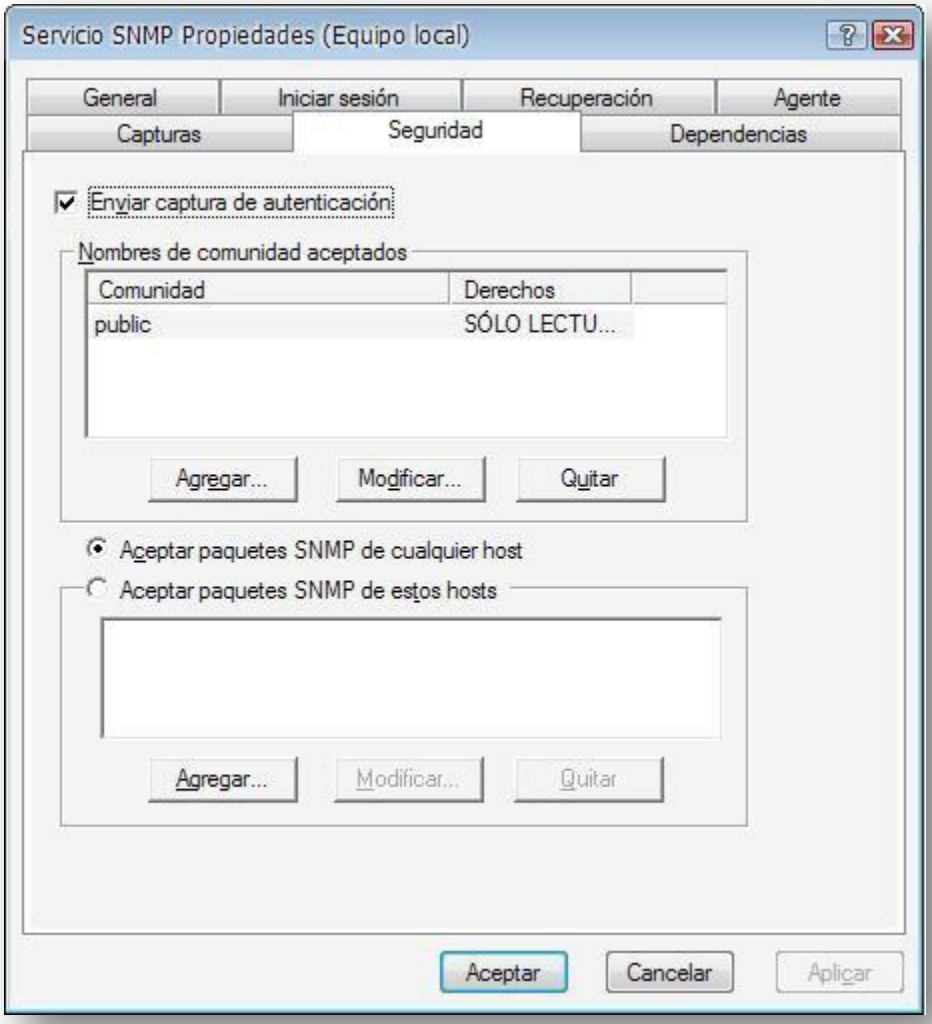



**Figura 65. Configuración del agente SNMP en Windows**

# 14.10. ANEXO X. Configuración de las cámaras de videovigilancia Sony

Una vez introduzcamos en la barra de direcciones del explorador Web la dirección IP de la cámara de videovigilancia se nos presentará el menú de entrada a la configuración:

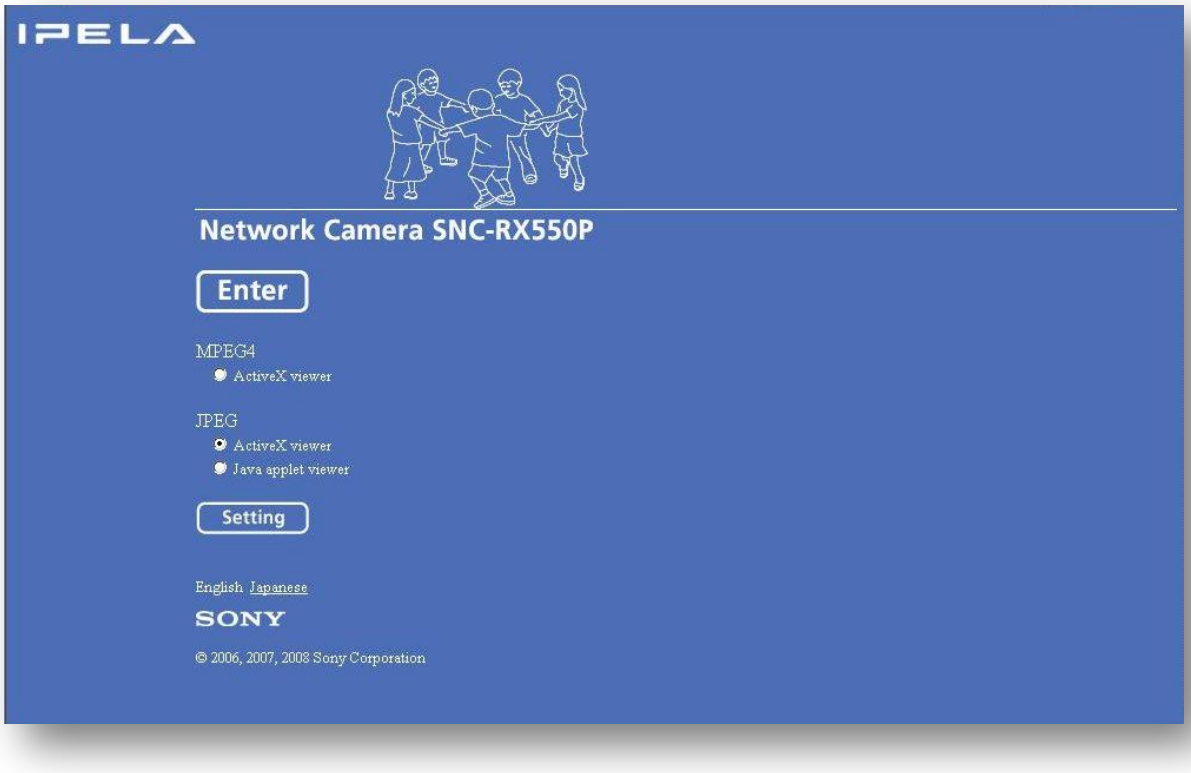



**Figura 66. Interfaz de acceso a la monitorización y configuración de una cámara Sony**

Pulsando sobre el botón *Enter*, accederemos a la funcionalidad para la monitorización de la imagen tomada por la cámara en el momento actual (ver siguiente página):

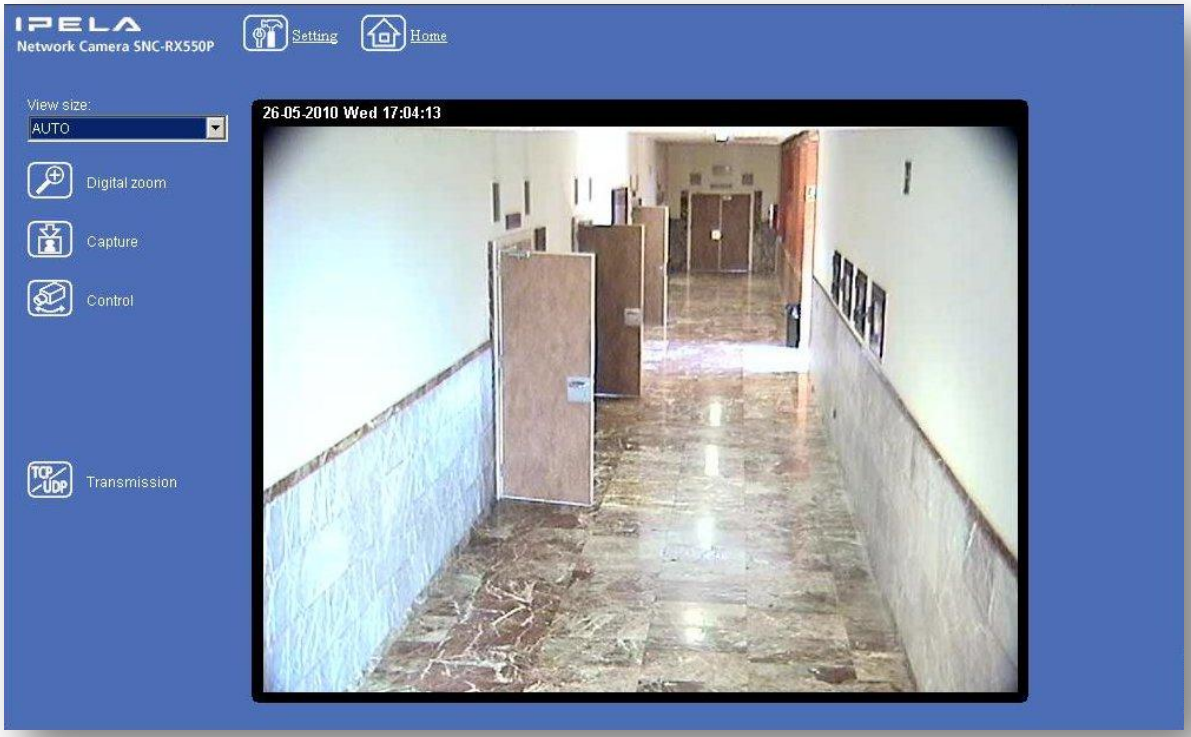



**Figura 67. Visor principal para monitorización de la imagen de una cámara Sony**

Dependiendo del modelo concreto de cámara, la interfaz web de monitorización ofrecerá distintas funcionalidades. Así, en el caso de la figura descrita, la imagen presentada ha sido captada por una cámara domo móvil, en concreto el modelo SNC-RX550P, por lo que es posible controlar el movimiento de la misma desde la propia interfaz de monitorización. Para una cámara fija esta funcionalidad no está permitida dada la propia limitación física de la cámara para realizar cualquier tipo de movimiento sobre su eje.

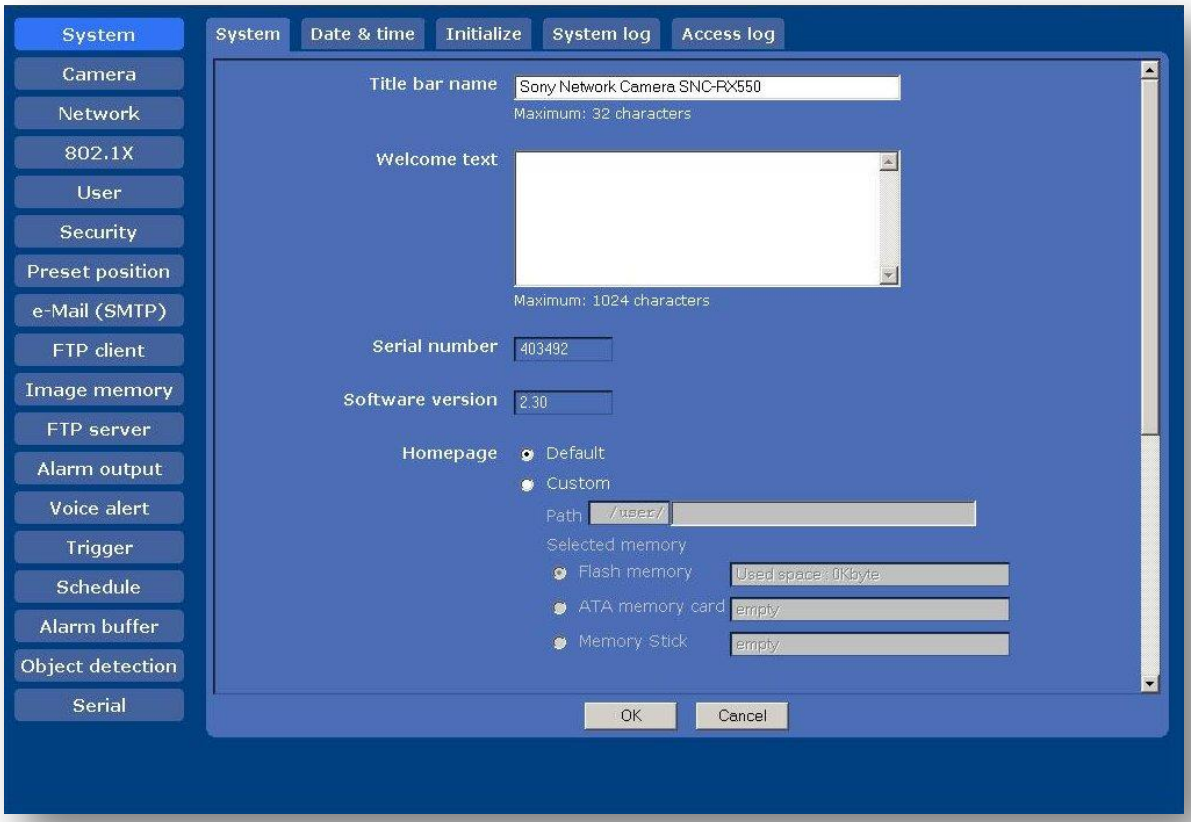



**Figura 68. Interfaz web para la configuración de una cámara Sony**

Habiendo iniciado sesión con el usuario autorizado, tendremos el menú reflejado en la figura anterior. Desde dicho menú administraremos las distintas opciones de configuración disponibles para un modelo concreto de cámara. Esas opciones están resumidas en la siguiente lista:

- **System**: este menú consta de varias fichas (*System, Date & Time, Initialize, System Log, Access log*). La primera de ellas, homónima al menú, permite asignar un nombre a la cámara así como un texto de bienvenida a mostrar cuando se inicie sesión en la misma. Igualmente, presenta el número de serie de la cámara y la versión de software instalada en la misma. Desde la ficha *System* también es posible cambiar la manera de operar del sensor de entrada y la página de inicio que se abrirá en el momento en que se introduzca la dirección IP de la cámara en la barra de direcciones del explorador Web. En la segunda ficha, ***Date & Time***, tal como su nombre indica, se establece el formato de fecha y hora que se mostrará en el visor principal de la cámara. El formato a seguir puede ser **yyyy-mm-dd hh:mm:ss**, **mm-dd-yyyy hh:mm:ss**, o **dd-mm-yyyy hh:mm:ss**. Asimismo, en esta ficha se mostrará la fecha y hora

establecidas en el equipo desde el cual se inicia sesión en la cámara, y se ofrecerán distintas opciones para establecer el día y la hora (sincronización con PC, configuración manual, sincronización vía servidor NTP). Otras funcionalidades que podemos encontrar en esta ficha se refieren a la presentación del identificador de la cámara y de la fecha de tal manera que aparezcan superpuestas en la imagen tomada. Desde la ficha ***Initialize*** se podrá reiniciar la cámara o restablecer su configuración a los parámetros por defecto asignados de fábrica. En ***System Log*** nos encontraremos el registro de los datos de actividad del software de la cámara, incluyendo datos que resultarán de utilidad en caso de producirse un problema en ella. Por último, en la ficha ***Access Log*** obtendremos el registro de acces.

- **Camera**: este menú, al igual que *System*, consta de varias fichas desde las cuales se configurará la imagen y el sonido de la cámara. En la primera ficha, ***Common,*** en el caso de la imagen, los parámetros a ajustar van desde el modo de imagen (*Field, Frame, Auto*), el color (monocromo o a color) hasta el zoom de la cámara (modos *Full* u *Optical*). En el menú de configuración del sonido de esa misma ficha se presentan las opciones para la conexión de un micrófono, la selección del códec de audio e incluso para la emisión de sonido desde el altavoz que estuviera conectado a la toma de salida de línea de la cámara. La segunda ficha, ***Picture***, permite ajustar el balance de blancos, el modo de exposición, el brillo, la saturación y el contraste. La ficha ***Day&Night*** ofrece un menú de selección para el modo "*Día/Noche*" (activado o desactivado) y el comportamiento del mismo a la hora de activarse en caso de que se decida habilitarlo. La siguiente ficha, ***Video codec***, se reserva para establecer los parámetros del códec de vídeo utilizado; *Single codec*, para elegir un códec entre JPEG, MPEG4 y H.264, y *Dual códec*, para utilizar simultáneamente los códecs JPEG y MPEG4. El resto de opciones de esta ficha son relativas a la calidad de imagen o al bitrate por citar algunas. Finalmente, en la ficha ***Streaming*** se ajustará la configuración de la cámara en términos de transmisión mediante monodifusión o multidifusión.
- **Network**: la función de este menú es establecer la conexión de red de la cámara. Al igual que los menús anteriores, se sigue una estructura de varias fichas. Desde la ficha ***Network*** se indica la dirección IP, la dirección MAC, la dirección de los servidores DNS, el nombre de host de la cámara para su transmisión en la red, y el puerto HTTP en el que escuchará ésta. La ficha ***Wireless***, se reserva para aquellos casos en los que se hace uso de una tarjeta de red LAN inalámbrica en la ranura de tarjetas de la propia cámara. En ese caso, los parámetros a configurar son los mismos que en la ficha *Network*, añadiéndose otras opciones para modificar la configuración de la antena, el identificador de la red inalámbrica y la seguridad de dicha red. La siguiente ficha, ***Dynamic IP address notification***, está creada para que, en el caso de seleccionar la obtención de un IP de forma automática (DHCP activado), se envíen notificaciones de finalización de la configuración de la red mediante el

protocolo SMTP o HTTP. Por último, la ficha ***SSL*** está ideada para ajustar la comunicación *SSL*[26] entre el equipo cliente y la cámara.

- **802.1X**: este menú posibilita configurar la autenticación basada en puerto, por cable o inalámbrica, de acuerdo con el estándar 802.1X. Para establecer una red 802.1X se debe configurar el autentificador, el punto de acceso, el servidor de autentificación y otros elementos a través de las distintas fichas de las que consta este menú. En la ficha ***Common*** se realiza la configuración básica de autenticación 802.1X; en la ficha ***Cliente certificate*** se permite importar un certificado de cliente en la cámara o exportar una solicitud de certificado; en la ficha ***CA certificate*** se ofrece la funcionalidad de importación en la cámara de un certificado de *CA*[27] de confianza (certificado de servidor o certificado de ruta). Se puede importar en la cámara hasta un total de cuatro certificados de este tipo, admitiéndose únicamente el formato *PEM*[28].
- **User**: en el menú *User* se establecen los nombres de usuario y contraseñas de Administrador. Se puede introducir, además del usuario *Administrador*, un total de hasta 9 tipos de usuarios con sus respectivos privilegios.
- **Security**: se hará uso de este menú cuando se desee restringir el conjunto de equipos que pueden tener acceso a la cámara. Dicho acceso se podrá filtrar a nivel de direcciones de red y valores de máscara de subred hasta un total de 10 de ellas.
- **Preset Position**: en aquellas cámaras en las que físicamente sea posible (cámara móvil), este menú permite guardar las posiciones de barrido horizontal, vertical y zoom así como establecer recorridos (acciones programadas de la cámara). Se tienen dos fichas; la primera de ellas, ***Position***, presenta las opciones necesarias para configurar hasta 16 posiciones de cámara (posiciones de barrido horizontal, vertical y zoom). Cada una de esas posiciones de barrido está caracterizada por un número, un nombre y las alarmas que tiene sincronizadas (*Entrada de sensor 1, Entrada de sensor 2* y *Detección de objetos*); en la segunda ficha, ***Tour***, se programan las 16 posiciones definidas en la ficha anterior y a las que la cámara se moverá secuencialmente (recorrido). Se pueden establecer, como máximo, 5 programas como recorridos. Las posibilidades de configuración abarcan desde el periodo durante el cual se activa el recorrido (en cualquier momento o según un programa establecido), el tiempo durante el cual la cámara permanece en cada posición preestablecida (entre 1 y 3600 segundos), la velocidad de movimiento de la cámara (desde 1 a 23, donde la velocidad es mayor cuanto mayor es el valor asignado) hasta la secuencia del recorrido.

[26] Secure Socket Layer
[27] Certificate Authority, autoridad de certificación
[28] Privacy Enhanced Mail

- **e-Mail (SMTP)**: mediante la función *e-Mail (SMTP)* se puede enviar un correo electrónico con archivos de imágenes adjuntas capturadas o en respuesta a la entrada de un sensor externo o bien a la función incorporada de detección de objetos. También es posible enviar el archivo de imagen de forma periódica. Siguiendo la estructura de los demás menús, nos encontramos con tres fichas. En primer lugar, la ficha ***Common*** nos va a permitir configurar la función de e-mail, activándola o bien deshabilitándola. Las opciones ofrecidas se refieren al nombre del servidor *SMTP*, al método de autenticación para el envío del correo electrónico, al nombre del servidor *POP*, al nombre y contraseña del usuario propietario de la cuenta de correo electrónico desde la que se envían los correos electrónicos, a la dirección de correo electrónico del destinatario y del Administrador y al asunto y cuerpo del mensaje que sería enviado. En la siguiente ficha, ***Alarm sending***, podremos vincular el envío de un mensaje de correo electrónico a la detección de una alarma mediante la entrada del sensor externo o mediante la función de detección de objetos. Se permite el envío de adjuntos en forma de archivo de imagen así como configurar la alarma a la cual estaría asociado el envío del mensaje. Por último, desde la ficha ***Periodical sending*** se podrá establecer que se envíe correo electrónico de forma periódica. Para ello, se indicará un intervalo y un periodo durante el cual estará activo el envío periódico.
- **FTP client**: utilizaremos este menú para configurar la captura y el envío de imágenes estáticas a un servidor FTP. Así, podremos enviar una imagen y un archivo de sonido grabados en respuesta a la entrada del sensor externo o a la función de detección de objetos, tal como hacíamos en el menú de envío de e-mail vía SMTP. En este menú disponemos de tres fichas al igual que en el menú anterior. ***Common*** presenta las opciones para la configuración del cliente y del servidor FTP. ***Alarm sending*** nos permitirá configurar la acción del cliente FTP ante una detección de alarma, y ***Periodical sending*** se reservará para el envío periódico de archivos de imagen JPEG al servidor FTP configurado.
- **Image memory**: mediante la función de memoria de imagen se puede grabar un archivo de imagen en diferentes ubicaciones, como puede ser dentro de la memoria incorporada en la cámara o en una tarjeta de memoria externa. También es posible grabar el archivo de imagen de forma periódica. Los archivos de imagen y sonido pueden buscarse o descargarse vía FTP en el equipo en el que se inicia sesión en la cámara. De manera similar a los dos menús anteriores, disponemos de tres fichas, ***Common, Alarm recording*** y ***Periodical recording*** desde las cuales podremos configurar, respectivamente, la función de memoria de imagen, el comportamiento de ésta asociado a una detección de alarma y el modo de grabación periódica. Las extensiones de permitidas para los archivos enviados o grabados mediante la función de imagen o la función de cliente FTP con ***".m4f", ".jpf"*** y ***".jpg".***

- **FTP server**: se utilizará este menú para configurar la función de servidor FTP que busca o descarga un archivo especificado de imagen y audio almacenado en una ubicación de memoria (memoria interna de la cámara o tarjeta externa).
- **Alarm output**: con la configuración de este menú se puede controlar la salida de alarma del puerto de E/S de la parte posterior de la cámara en respuesta a la detección de la alarma, al temporizador y a la función "*Día/Noche*". Las opciones permiten establecer el modo de funcionamiento (por detección de alarma, por temporizador o por función día/noche), la vinculación de la salida de alarma a la función de detección del objeto, la duración de la alarma y el tiempo durante el cual está activada la alarma en caso de haber seleccionado la detección de alarma como modo de funcionamiento.
- **Voice alert**: desde este menú se establece la función de alerta de voz para emitir sonido desde la toma de salida de línea de la cámara en caso de detectar una alarma a través de la entrada del sensor o de la función de detección de objetos. El menú consta de 3 fichas, ***Voice alert 1, Voice alert 2*** y ***Voice alert 3***. En todas ellas se configura el nombre del archivo de sonido grabado en la cámara, la vinculación de la alerta de voz a la entrada del sensor o a la detección de objetos, el número de repeticiones para la reproducción del sonido (de 1 a 3), la alarma que se vincularía con la función de alerta de voz y el tiempo durante el cual estaría activa la detección de dicha alarma.
- **Trigger**: en este menú se seleccionan las actividades que se realizarán cuando se acceda a la sección correspondiente en el visor principal.
- **Schedule**: con el menú *Schedule* se establece el periodo efectivo en los menús que soporten este parámetro.
- **Alarm buffer**: en este punto se establece la imagen y el sonido pre-alarma (la imagen y sonido anteriores a la detección de la alarma) y la imagen y el sonido post-alarma. Las opciones nos permiten comprobar el códec de vídeo seleccionado y la capacidad máxima de grabación del búfer de alarma en la configuración actual de la cámara. Desde aquí se podrá personalizar el tiempo de grabación para la imagen y el sonido pre-alarma así como para la imagen y el sonido post-alarma.
- **Object detection**: mediante este menú se indican las condiciones de activación de las funciones "*Moving object detection*" y "*Unattended object detection*". La detección de objetos en movimiento detecta objetos que se mueven en la imagen de la cámara y emite una alarma. La detección de objetos abandonados (*unattended*) detecta las diferencias entre la imagen de fondo grabada previamente y la imagen actual, emitiendo una alarma cuando la diferencia continúa reconociéndose durante más de un periodo especificado.

- **Serial:** con este menú se pueden introducir datos en la cámara desde un equipo a través de una red y enviarlos a la interfaz serie externa para controlar un dispositivo periférico. También se permite la operación inversa, es decir, la introducción de datos procedentes de un dispositivo periférico mediante una interfaz serie externa. Desde aquí configuraremos el modo de transmisión y recepción de datos (a elegir entre *TCP* y *VISCA*[29]) así como la paridad, la longitud en caracteres, el bit de parada y la velocidad en baudios correspondientes a la interfaz serie del dispositivo periférico.

[29] Video System Control Architecture, estándar del fabricante Sony

# 14.11. ANEXO XI. Script de rotación de logs MySQL.

```
/var/log/mysql/mysql.log  /var/log/mysql/mysql-slow.log
{
        daily
        rotate 7
        missingok
        create 640 mysql adm
        compress
        sharedscripts
        postrotate
            test -x /usr/bin/mysqladmin || exit 0

            # If this fails, check debian.conf!
            MYADMIN="/usr/bin/mysqladmin --defaults-file=/etc/mysql/debian.cnf"
            if [ -z "`$MYADMIN ping 2>/dev/null`" ]; then
              if ps cax | grep -q mysqld; then
                exit 1
              fi
            else
              $MYADMIN flush-logs
            fi
        endscript
}
```

Observando este fichero, vemos que los ficheros de log que se están rotando corresponden, respectivamente, al registro de consultas (**mysql.log**) y al registro de consultas lentas (**mysql-slow.log**).

El parámetro "**daily**" indica que la rotación de los ficheros se hace diariamente; "**rótate 7**" significa que los ficheros de log implicados se rotarán durante un total de 7 días; "**missingok**" es una opción utilizada para que, en caso de que el fichero de log no se encuentre, se intente rotar el siguiente fichero de log sin mostrar un mensaje de error de que no se encontró el anterior; "**create 640 mysql adm**" indica que, inmediatamente después de la rotación, se crea el fichero de log con un modo (640), propietario (mysql) y grupo (adm); "**compress**" hará que las versiones rotadas se compriman con **gzip** por defecto; con "**sharedscripts**" especificado, si se indican scripts previos y posteriores a la rotación (*prescript y postscript*), éstos se ejecutarán una única vez sin importar el número de ficheros de Logs que se ajusten al patrón "*/var/log/mysql/mysql.log /var/log/mysql/mysql-slow.log*" y, finalmente, todas las

sentencias escritas entre "**postrotate**" y "**endscript**" serán las acciones que se ejecutarán una vez el fichero sea rotado. En este caso, si el servidor MySQL se está ejecutando, se hará una limpieza de sus Logs a través del comando "**FLUSH LOGS**".

# 14.12. ANEXO XII. Script de backup de la base de datos MySQL de Zabbix

```
#!/bin/bash

fecha=`date +%Y%m%d-%H%M`

/usr/bin/mysqldump --defaults-extra-file=/etc/mysql/backupcredentials.cnf --add-drop-table --add-locks --extended-insert --single-transaction --quick 'zabbix' | bzip2 > "/BackupDBzabbix/zabbixDBdump$fecha.sql".bz2
```

Este script se encarga de crear un fichero de backup a través de mysqldump con las opciones vistas en la sección 8.1 para comprimirlo en un fichero con formato "bz2." que ocupe menos espacio en su ubicación final.

# 14.13. ANEXO XIII. Soft States/Hard States en Nagios

Nagios funciona comprobando si un *host* o un *service* concreto funcionan correctamente y almacena su estado. Como dicho estado toma un valor de cuatro posibles, es crucial que refleje fielmente cuál es el estado actual. Con objeto de evitar detectar falsas alarmas o problemas temporales, Nagios utiliza esta funcionalidad de *soft states* y *hard states* para describir cuál es el estado actual de un *host* o de un *service*.

Básicamente, esa funcionalidad consiste en comprobar varias veces el estado del *host* o del *service* para asegurarse de que su estado actual realmente es el que se obtiene al hacer la comprobación. Cuando el estado es desconocido o diferente del anterior, Nagios volverá a ejecutar un check sobre el *host* o *service* para verificar si el cambio en el estado es persistente. El nuevo valor del estado se considera el *soft state* y si, tras sucesivas comprobaciones, éste no cambia, se considera que el valor es permanente y pasa a ser el *hard state*. El número de comprobaciones que Nagios efectúa es configurable por el usuario.

Sirva como ejemplo el mostrado en la siguiente figura (extraído del libro '*Learning Nagios 3.0*'):

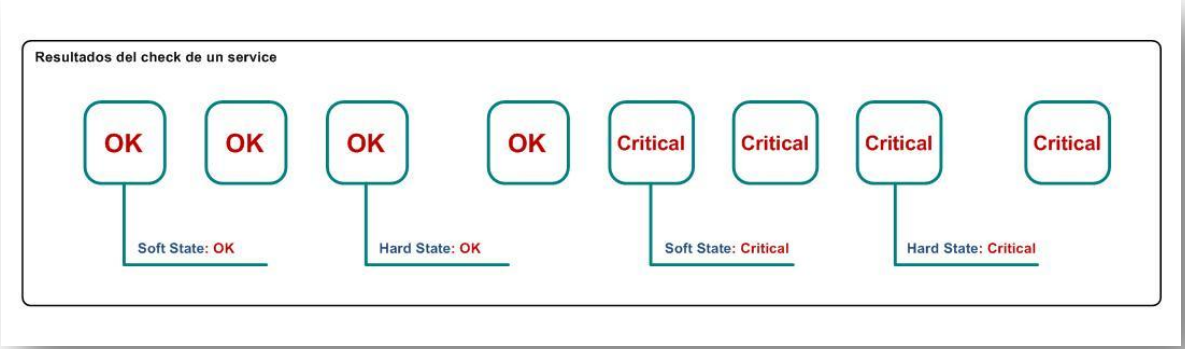



**Figura 69. Ejemplo de soft states/hard states en Nagios**

En este ejemplo, el número de comprobaciones que Nagios hará para validar el estado de un *service* es igual a 3. Inicialmente, el estado del service es "OK" y consideramos que ha cambiado con respecto al valor que tenía anteriormente. Entonces, "OK" será el soft state. Nagios efectúa una segunda comprobación y observa que el valor "OK" persiste, por lo que pasa a ejecutar una tercera iteración, obteniendo nuevamente el valor "OK", por lo que interpreta que realmente ese es el valor corespondiente al estado del servicio. En ese instante, "OK" pasa a ser hard state, de modo que Nagios notifica que el estado del servicio es "OK".

Nagios sigue comprobando el valor del servicio y nuevamente obtiene el valor "OK", pero como ya ha llegado al límite de comprobaciones (3) y el valor anterior también era "OK", no se produce cambio alguno en los soft/hard states.

Sin embargo, en la siguiente comprobación, el valor del service cambia a "Critical". Como es diferente al anterior estado ("OK"), Nagios establece un nuevo soft state y ahora lo hace con valor "Critical". Nagios vuelve a comprobar de nuevo el service y obtiene el mismo valor "Critical", por lo que el soft state no cambia y pasa a realizar la tercera comprobación. En ésta, el valor es "Critical", así que el nuevo hard state pasa a ser "Critical" y Nagios informará que ese es el estado del servicio.

[Fin del documento]